IARIA

INFOCOMP 2011

The First International Conference on Advanced Communications and Computation

ISBN: 978-1-61208-161-8

October 23-29, 2011

Barcelona, Spain

INFOCOMP 2011 Editors

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

Wolfgang Christmann, christmann informationstechnik + medien GmbH & Co. KG

Subhash Saini, NASA, USA

Malgorzata Pankowska, University of Economics, Katowice, Poland

INFOCOMP 2011

Foreword

The First International Conference on Advanced Communications and Computation [INFOCOMP 2011], held between October 23 and 29, 2011 in Barcelona, Spain, inaugurated a series of events dedicated to advanced communications and computing aspects, covering academic and industrial achievements and visions.

The diversity of semantics of data, context gathering and processing led to complex mechanisms for applications requiring special communication and computation support in terms of volume of data, processing speed, context variety, etc. The new computation paradigms and communications technologies are now driven by the needs for fast processing and requirements from data-intensive applications and domain-oriented applications (medicine, geoinformatics, climatology, remote learning, education, large scale digital libraries, social networks, etc.). Mobility, ubiquity, multicast, multi-access networks, data centers, cloud computing are now forming the spectrum of de factor approaches in response to the diversity of user demands and applications. In parallel, measurements control and management (self-management) of such environments evolved to deal with new complex situations.

We take here the opportunity to warmly thank all the members of the INFOCOMP 2011 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to INFOCOMP 2011. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the INFOCOMP 2011 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that INFOCOMP 2011 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the areas of communications and computations.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the historic charm of Barcelona, Spain.

INFOCOMP 2011 Chairs:

General Chair
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

Advisory Chairs
Hans-Joachim Bungartz, Technische Universität München (TUM), Germany
Laura Carrington, University of California at San Diego (UCSD) / San Diego Supercomputer Center (SDSC), USA
Petre Dini, Concordia University, Canada / IARIA, USA
Erik Elmroth, Department of Computing Science and HPC2N, Umeå University, Sweden
Sik Lee, Supercomputing Center KISTI, Korea Institute of Science and Technology Information (KISTI), Korea
Subhash Saini, NASA, USA

Academia Chairs

Bruce Berriman, California Institute of Technology, USA
Evgenia N. Cheremisina, Dubna International University, Russia / Academic of Russian Academy of Natural Sciences, Moscow, Russia
Malgorzata Pankowska, University of Economics, Katowice, Poland

Research Institute Liaison Chairs

Kei Davis, Los Alamos National Laboratory / Computer, Computational, and Statistical Sciences Division, USA
Daniel S. Katz, Computation Institute, University of Chicago & Argonne National Laboratory, USA
Harald Kornmayer, Cooperative State University of Baden-Wuerttemberg, Mannheim, Germany
Ivor Spence, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Northern Ireland, Research for High Performance and Distributed Computing / Queen's University Belfast, UK

Industry Chairs

Wolfgang Christmann, christmann informationstechnik + medien GmbH & Co. KG
William R. Claycomb, CERT Insider Threat Research Center/Carnegie Mellon University, USA
Edgar Leon, IBM Research, Austin, USA
Hans-Günther Müller, SGI Silicon Graphics - Göttingen, Germany
INFOCOMP Special Area Chairs on Large Scale and Fast Computation
Christopher Jordan, The University of Texas at Austin, USA
Walter Lioen, SARA Computing and Networking Services Amsterdam, The Netherlands

Special Area Chairs on Networks and Communications

Noelia Correia, University of the Algarve, Portugal
Wolfgang Hommel, Leibniz Supercomputing Centre - Munich, Germany
George Michelogiannakis, Stanford University, USA

Special Area Chairs on Advanced Applications

Kurt Schneider, Fachgebiet Software Engineering, Leibniz Universität Hannover, Germany

Special Area Chairs on Evaluation Context

Dominic Eschweiler, Jülich Supercomputing Centre, Germany
Philipp Kremer, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Oberpfaffenhofen, Germany

INFOCOMP 2011

Committee

INFOCOMP General Chair

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

INFOCOMP Advisory Chairs

Hans-Joachim Bungartz, Technische Universität München (TUM), Germany
Laura Carrington, University of California at San Diego (UCSD) / San Diego Supercomputer Center (SDSC), USA
Petre Dini, Concordia University, Canada / IARIA, USA
Erik Elmroth, Department of Computing Science and HPC2N, Umeå University, Sweden
Sik Lee, Supercomputing Center KISTI, Korea Institute of Science and Technology Information (KISTI), Korea
Subhash Saini, NASA, USA

INFOCOMP Academia Chairs

Bruce Berriman, California Institute of Technology, USA
Evgenia N. Cheremisina, Dubna International University, Russia / Academic of Russian Academy of Natural Sciences, Moscow, Russia
Malgorzata Pankowska, University of Economics, Katowice, Poland

INFOCOMP Research Institute Liaison Chairs

Kei Davis, Los Alamos National Laboratory / Computer, Computational, and Statistical Sciences Division, USA
Daniel S. Katz, Computation Institute, University of Chicago & Argonne National Laboratory, USA
Harald Kornmayer, Cooperative State University of Baden-Wuerttemberg, Mannheim, Germany
Ivor Spence, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Northern Ireland, Research for High Performance and Distributed Computing / Queen's University Belfast, UK

INFOCOMP Industry Chairs

Wolfgang Christmann, christmann informationstechnik + medien GmbH & Co. KG
William R. Claycomb, CERT Insider Threat Research Center/Carnegie Mellon University, USA
Edgar Leon, IBM Research, Austin, USA
Hans-Günther Müller, SGI Silicon Graphics - Göttingen, Germany

INFOCOMP Special Area Chairs on Large Scale and Fast Computation

Christopher Jordan, The University of Texas at Austin, USA
Walter Lioen, SARA Computing and Networking Services Amsterdam, The Netherlands

INFOCOMP Special Area Chairs on Networks and Communications

Noelia Correia, University of the Algarve, Portugal
Wolfgang Hommel, Leibniz Supercomputing Centre - Munich, Germany
George Michelogiannakis, Stanford University, USA

INFOCOMP Special Area Chairs on Advanced Applications

Kurt Schneider, Fachgebiet Software Engineering, Leibniz Universität Hannover, Germany

INFOCOMP Special Area Chairs on Evaluation Context

Dominic Eschweiler, Jülich Supercomputing Centre, Germany
Philipp Kremer, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Oberpfaffenhofen, Germany

INFOCOMP 2011 Technical Program Committee

Riccardo Albertoni, CNR-IMATI-GE, Italy
Tulin Atmaca, Institut Telecom & Management SudParis, France
Bernhard Bandow, Thomas-Mann-Schule Lübeck, Germany
Bruce Berriman, California Institute of Technology, USA
Stefano Bistarelli, Università di Perugia / Istituto di Informatica e Telematica - C.N.R. - Pisa, Italy
Jonathan M. Blackledge, College of Engineering and the Built Environment / Dublin Institute of Technology (DIT), Ireland
Sara Bouchenak, Grenoble University / INRIA, France
Hans-Joachim Bungartz, TUM - Garching, Germany
Elena Camossi, European Commission / Joint Research Centre, IPSC - Ispra, Italia
Laura Carrington, University of California at San Diego (UCSD) / San Diego Supercomputer Center (SDSC), USA
Wojciech Cellary, Poznan University of Economics, Poland
Evgenia N. Cheremisina, Dubna International University / Academic of Russian Academy of Natural Sciences - Moscow, Russia
Wolfgang Christmann, christmann informationstechnik + medien GmbH & Co. KG - Ilsede, Germany
William R. Claycomb, CERT Insider Threat Research Center/Carnegie Mellon University, USA
Marco Comuzzi, Eindhoven University of Technology, The Netherlands
Noelia Correia, University of Algarve, Portugal
Kei Davis, Los Alamos National Laboratory / Computer, Computational, and Statistical Sciences Division, USA
Harrison (Nick) Eiteljorg, II, Center for the Study of Architecture (CSA), USA
Erik Elmroth, Umeå University / High Performance Computing Center North (HPC2N), Sweden
Dominic Eschweiler, Jülich Supercomputing Centre, Germany
Irina Fedulova, IBM Russian Systems and Technology Laboratory - Moscow, Russia
Birgit Frida Stefanie Gersbeck-Schierholz, Leibniz Universität Hannover (LUH) / Certification Authority University of Hannover (UH-CA), Germany
Franca Giannini, IMATI - Consiglio Nazionale delle Ricerche - Genova, Italy
Richard Gunstone, Bournemouth University, UK
Jean Hennebert, University of Applied Sciences HES-SO - Sierre, Switzerland
Vincent Heuveline, KIT, University of the State of Baden-Wuerttemberg / National Large-scale Research Center of the Helmholtz Association, Germany
Iman Hong, Soongsil University - Seoul, Korea
Wolfgang Hommel, Leibniz Supercomputing Centre - Munich, Germany
Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek - Hannover, Germany
Liviu Gr. Ixaru, National Institute of Physics and Nuclear Engineering - Bucharest / Academy of Romanian Scientists, Romania
Christopher Jordan, The University of Texas at Austin, USA

Daniel S. Katz, Computation Institute, University of Chicago & Argonne National Laboratory, USA
Abdelmajid Khelil, Darmstadt University of Technology, Germany
Christos Kloukinas, City University London, UK
Alexander Knapp, Universität Augsburg, Germany
Harald Kornmayer, Cooperative State University of Baden-Wuerttemberg - Mannheim, Germany
Philipp Kremer, German Aerospace Center (DLR)/Institute of Robotics and Mechatronics - Oberpfaffenhofen, Germany
Sik Lee, Supercomputing Center / Korea Institute of Science and Technology Information (KISTI), Korea
Edgar Leon, IBM Research - Austin, USA
Walter Lioen, SARA Computing and Networking Services - Amsterdam, The Netherlands
Patrick Ludwig, Institute of Theoretical Physics and Astrophysics, Christian-Albrechts-Universität zu Kiel, Germany
Paulo Martins, Chaminade University, USA
Ludek Matyska, ICS, Masaryk University /Supercomputing Centre Brno (SCB), Czech Republic
George Michelogiannakis, Stanford University, USA
Hans-Günther Müller, SGI Silicon Graphics - Göttingen, Germany
Lena Noack, Joint Planetary Interior Physics Research Group, University Münster / IfP DLR - Berlin, Germany
Stephan Olbrich, University of Hamburg/ Regional Computer Center, Germany
Martin M. Ossowski, Rice University - Houston, USA
Malgorzata Pankowska, University of Economics - Katowice, Poland
Nikos Parlavantzas, IRISA / Campus Universitaire de Beaulieu - Rennes, France
Giuseppe Patanè, CNR-IMATI, Italy
Ana-Catalina Plesa, Joint Planetary Interior Physics Research Group, University Münster / IfP DLR - Berlin, Germany
Ivan Rodero, Rutgers University - Piscataway, USA
Claus-Peter Rückemann, WWU Münster / Leibniz Universität Hannover / HLRN, Germany
H. Birali Runesha, University of Minnesota - Minneapolis / Supercomputing Institute for Advanced Computational Research, USA
Subhash Saini, NASA, USA
Kurt Schneider, Fachgebiet Software Engineering, Leibniz Universität Hannover, Germany
Benjamin Schütze, Institute for Legal Informatics, Leibniz Universität Hannover, Germany
Dongwan Shin, New Mexico Tech University, USA
Theodore E. Simos, King Saud University - Riyadh, Saudi Arabia / University of Peloponnese - Tripolis, Greece
Happy Sithole, Center for High Performance Computing (CHPC), Cape Town, South Africa
Ivor Spence, Research for High Performance and Distributed Computing / Queen's University Belfast, UK
Florian Stegmaier, University of Passau, Germany
Monika Steinberg, Leibniz Universität Hannover / SRA, Germany
Rahim Tafazolli, Center for Communication Systems Research (CCSR), University of Surrey - Guildford, UK
Rob Thacker, Saint Mary's University - Halifax, Canada
Francesco Tiezzi, Università degli Studi di Firenze, Italy
Dan Tulpan, National Research Council of Canada / Université de Moncton, New Brunswick, Canada
Ravi Vadapalli, High Performance Computing Center (HPCC) / Texas Tech University - Lubbock, USA
Eloisa Vargiu, University of Cagliari, Italy
Edward Walker, Whitworth University, USA
Cho-Li Wang, University of Hong Kong, Hong Kong
Stephen White, The University of Huddersfield, UK
David Isaac Wolinsky, University of Florida, USA
Weijia Xu, The University of Texas at Austin, USA
Weikuan Yu, Auburn University, USA

Copyright Information

Table of Contents

Implementation of Integrated Systems and Resources for Information and Computing

Claus-Peter Rückemann
Leibniz Universität Hannover,
Westfälische Wilhelms-Universität Münster (WWU),
North-German Supercomputing Alliance (HLRN), Germany
Email: `ruckema@uni-muenster.de`

***Abstract*—This paper presents the practical results from the real-life implementation of interactive complex systems in specialised High End Computing environments. The successful implementation has been made possible using a new concept of higher-level data structures for dynamical applications and configuration of the resources. The discussion shows how an implementation of integrated information systems, compute and storage resources has been achieved. The implementation uses techniques ensuring to create a flexible way for communication with complex information and computing systems. Besides Inter-Process Communication these are mainly Object Envelopes for object handling and Compute Envelopes for computation objects. These algorithms provide means for generic data processing and flexible information exchange. Targeting mission critical environments, the interfaces can embed instruction information, validation and verification methods. The application covers challenges of collaborative implementation, legal, and security issues with these processes. The focus is on integrating information and computing systems, Distributed and High Performance Computing (HPC) resources, for use in natural sciences disciplines and scientific information systems. Implementing higher-level data structure frameworks for dynamical applications and resources configuration has led to scalable and modular integrated public / commercial information system components.**

Keywords*–Integrated Systems; Information Systems; Computing Systems; Geosciences; High Performance Computing

I. Introduction

In High Performance Computing (HPC) supercomputers, that means computer systems at the upper performance limit of current technical capabilities for computing, are employed to solve challenging scientific problems. In consequence there is no general or common architecture and configuration for HPC resources as in the lower parts of the performance pyramid.

Within the last decades a large number of implementations of information systems, computing and storage systems and other resources have been created. Nearly all of these implementations lack features for extending information systems with the various resources available. Thus, the integration could be opening advances using larger resources, interactive processing, and reducing time consumption for assigned tasks. Most of these applications and resources are very complex, standalone systems and used that way, neglecting that for many sophisticated use cases conjoint applications are desirable.

The next generation of systems neccessary for providing profound means for communication and computation will have to gather methods evolved by active interdisciplinary interchange, grown with the requirements of the last decades: The information and computing disciplines need means for in praxi collaboration from disciplines, structural sciences, computer science, computing science, and information science. Examples are computing intensive interactive environmental monitoring and information systems or simulation supported dynamical geoinformation systems. In this manner an efficient development and operation can be put into practice for making interactive use of systems with tenthousands of thousands of nodes, ten to hundred thousands of compute cores and hundred thousands to millions of system parts like memory bars and hard disk drives. Methodological sciences means sciences of developing methods for using resources and techniques for gathering new scientific insights. For years, “methodological sciences” or more precise “methodological techniques” have been commonly propagated to solve the problems of High End Computing. It has been commonly experienced that this is not true as there are no applicatory results regarding the essential requirements of complex and integrated high end systems. The available “methods and techniques” is not what we can use for supercomputing where every application and system architecture is different. Up to now this difference is implicit with common tender and operation strategy for various efficiency and economical reasons.

The experiences with integrated systems have been compiled in various projects over the last years [1], [2]. Legal issues and object security have been discussed at the CYBERLAWS 2010 and 2011 conferences. The architecture of the framework and suitable components used, have been tested by implementing various integrated systems. The following sections show components of an integrated geoscientific information and computing systems developed in one of these case studies that can be used for environmental monitoring or feeding expert systems. None of the participating industry and scientific parties can or will create one single application from the components discussed here. The goal is to enable the necessary operation, nevertheless these are separate components. For the last years practical solutions to various requirements for communication requests have been implemented in a number of projects and case studies using various resources [3], [4], [5], [6], [7]. The most important communication facilities for integrated information and computing systems are:

- Communication requests with applications (example: Inter-Process Communication, IPC).
- Storage object requests (framework example: Object Envelopes, OEN).
- Compute requests (framework example: Compute Envelopes, CEN).

Based on these components an integrated solution has been built, for use with local HPC resources supported by distributed information and compute resources. From the point of view of resources providers and integrators of HPC resources it would make very little sense to describe the application components here. Applications details have been published for several components before. For the core issues the conceptional results are by far the most important.

This paper is organised as follows. Section two presents motivation, challenges and complexity with the implementation. Sections three and four describe the prerequisites and basic resources configuration for the implementation. Sections five and six show the components and dependencies for integrated systems and resources. Section seven discusses the time dependence of the integrated solutions. Section eight describes the system implementation, and Section nine does the evaluation. Sections ten and eleven summarise the lessons learned, conclusions, and future work.

II. Motivation

With the implementation use cases for Information Systems the suitability of Distributed and High Performance Computing resources have been studied. These use cases have focus on event triggered communication, dynamical cartography, compute and storage resources. The goal has been, to bring together the features and the experiences for an integrated information and computing system. An example that has been implemented is a spatial information system with hundreds of thousands of ad-hoc simulations of interest. Within these interactive systems as many “next informations of interest” as possible can be dynamically calculated in parallel, near real-time, in order to be of any practical use. In the following passages we will show an environmental component exactly using this implementation for many thousands of points of interest.

Due to the complexity of integrated information and computing systems, we have applied meta-instructions and signatures for algorithms and interfaces. For these cases, envelopes and IPC has been used to provide a unique event and process triggered interface for event, computing, and storage access.

III. System prerequisites

For implementation and testing a suitable system architecture and hardware had been neccessary. A single local system had to fulfill the following minimal criteria:

- Capacity for more than 5000 subjobs per job.
- At least one compute core available per subjob.
- Interactive batch system.
- No distributed compute and storage resources.
- Fast separate InfiniBand networks for compute and IO.
- Highly performant parallel filesystem.
- Available for being fully configurable.

A system provided being fully configurable means especially configuration of hardware, network, operating system, middleware, scheduling, batch system. At this size this normally involves a time interval of at least three to six months.

It should be obvious that there are not many installations of some reasonable size and complexity that could be provided, configured and operated that way if in parallel to normal operation and production.

The available HPC and distributed resources at ZIV and HLRN as well as commercially provided High End Computing installations have been sufficient to fulfill all the necessary criteria.

IV. Basic resources configuration

Elementary operating system components on the resources involved are: AIX, Solaris, and various Linux distributions (SLES, Scientific Linux). Elementary middlewares, protocols, and accounting systems used for the integrated components are: Globus Toolkit, SGAS, DGAS. Unicore, SAGA, SOAP, and many others can be integrated, too. For communication and parallelisation MPI (Open-MPI [8], MPI from SGI, Intel, HP, IBM), OpenMP, MPICH, MVAPICH and other methods have been used along with IPC regarding to the type of operation and optimisation of algorithms needed. For the scheduling and batch systems the resources used Moab, Torque, LoadLeveler, and SGE.

All these “tools” are only middleware components, protocols, interfaces or isolated applications. They are certainly used on the system resources but they cannot integrate anything, not on the disciplines/application level, not on the services level, not on the resources level. So we want to concentrate on the important high-level issues for the further advanced view of components.

V. Components

Using the following concepts, we can, mostly for any system, implement:

- Application communication via IPC.
- Application triggering on events.
- Storage object requests based on envelopes.

- Compute requests based on envelopes.

For demonstration and studies flexible and open Active Source Information System components have been used for maximum transparency. This allows OO-support (object, element) on application level as well as multi-system support. Listing 1 shows a simple example for application communication with framework-internal and external applications (Inter-Process Communication, IPC).

```
 catch {
  send {rasmol #1} "$what"
 }
```

Listing 1. Application communication (IPC).

This is self-descriptive Tcl syntax. In this case the IPC send is starting a molecular graphics visualisation tool and catching messages for further analysis by the components.

Listing 2 shows an example of how the communication triggering can be linked to application components.

```
text 450.0 535.0 -tags {itemtext relictrotatex} -fill
 yellow -text "Rotate␣x" -justify center
...
$w bind relictrotatex <Button-1> {sendAllRasMol {rotate
  x 10}}
$w bind relictballsandsticks <Button-1> {sendAllRasMol
 {spacefill 100}}
$w bind relictwhitebg <Button-1> {sendAllRasMol {set
 background white}}
$w bind relictzoom100 <Button-1> {sendAllRasMol {zoom
 100}}
```

Listing 2. Application component triggering.

Tcl language objects like text carry tag names (relictrotatex) and dynamical events like Button events are dynamically assigned and a user defined subroutine sendAllRasMol is executed, triggering parallel visualisation. Storage object requests for distributed resources can be done via OEN. Listing 3 shows a small example for a generic OEN file.

```
<ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
<Object>
<Filename>GIS_Case_Study_20090804.jpg</Filename>
<Md5sum>...</Md5sum>
<Sha1sum>...</Sha1sum>
<DateCreated>2010-08-01:221114</DateCreated>
<DateModified>2010-08-01:222029</DateModified>
<ID>...</ID><CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><ContentData>...</ContentData></Content>
</Object>
</ObjectEnvelope>
```

Listing 3. Storage object request (OEN).

OEN are containing element structures for handling and embedding data and information, like Filename and Content. An end-user public client application may be implemented via a browser plugin, based on appropriate services. With OEN instructions embedded in envelopes, for example as XML-based element structure representation, content can be handled as content-stream or as content-reference. Algorithms can respect any meta-data for objects and handle different object and file formats while staying transparent and portable. Using the content features the original documents can stay unmodified. The way this will have to be implemented for different use cases depends on the situation, and in many cases on the size and number of data objects. but the hierarchical structured meta data is uniform and easily parsable. Further it supports signed object elements (Signature), validation and verification via Public Key Infrastructure (PKI) and is usable with sources and binaries like Active Source. Compute requests for distributed resources are handled via CEN interfaces. Listing 4 shows a generic CEN file with embedded compute instructions. Content can be handled as content-stream or as content-reference (Content, ContentReference). Compute instruction sets are self-descriptive and can be preconfigured to the local compute environment.

```
<ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
<Instruction>
<Filename>Processing_Batch_GIS612.pbs</Filename>
<Md5sum>...</Md5sum>
<Sha1sum>...</Sha1sum>
<Sha512sum>...</Sha512sum>
<DateCreated>2010-08-01:201057</DateCreated>
<DateModified>2010-08-01:211804</DateModified>
<ID>...</ID>
<CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><DataReference>https://doi...</DataReference><
 /Content>
<Script><Pbs>
<Shell>#!/bin/bash</Shell>
<JobName>#PBS -N myjob</JobName>
<Oe>#PBS -j oe</Oe>
<Walltime>#PBS -l walltime=00:10:00</Walltime>
<NodesPpn>#PBS -l nodes=8:ppn=4</NodesPpn>
<Feature>#PBS -l feature=ice</Feature>
<Partition>#PBS -l partition=hannover</Partition>
<Accesspolicy>#PBS -l naccesspolicy=singlejob</
 Accesspolicy>
<Module>module load mpt</Module>
<Cd>cd $PBS_O_WORKDIR</Cd>
<Np>np=$(cat $PBS_NODEFILE | wc -l)</Np>
<Exec>mpiexec_mpt -np $np ./dyna.out 2>&1</Exec>
</Pbs></Script>
</Instruction>
</ComputeEnvelope>
```

Listing 4. Compute request (CEN).

In this case standard PBS batch instructions like walltime and nodes are used. The way this will have to be implemented for different use cases depends on the situation, an in many cases on the size and number of data objects. An important benefit of content-reference with high performant distributed or multicore resources is that references can be processed in parallel on these architectures. The number of physical parallel resources and the transfer capacities inside the network are limiting factors.

VI. Integrated systems with coupled resources

Figure 1 shows the applied integration of the information and communication systems with coupled computation resources, namely compute resources and storage resources. For integrating the features of information and communication systems with powerful compute resources and storage,

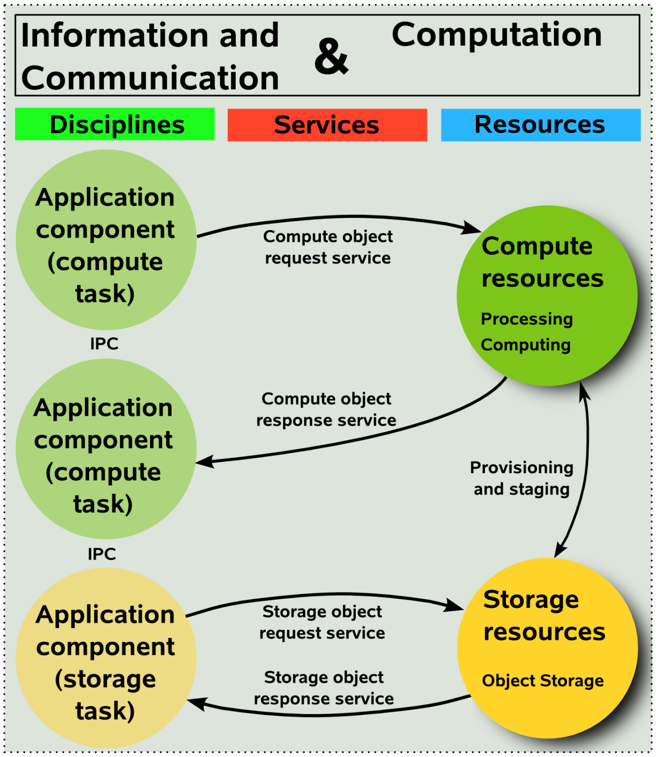

Information and Communication
&
Computation
Disciplines
Services
Resources
Application component (compute task)
IPC
Application component (compute task)
IPC
Application component (storage task)
Compute object request service
Compute object response service
Provisioning and staging
Storage object request service
Storage object response service
Compute resources
Processing
Computing
Storage resources
Object Storage

Figure 1. Integrated systems and resources.

it has been necessary to implement interfaces and software applications being able to efficiently use the benefits of High End Computing resources.

Following the results of the long-term case studies [9] three columns namely disciplines (as geosciences), services (as middleware and compute services), resources (computing and storage) had to be figured out for this scenario.

The discipline column shows application components with the state for a compute task and an application component with state for a storage task. Local tasks, ordinary communication between the applications without the need for external computing power, can as usual be done using a local service, for example using Inter-Process Communication (IPC).

Using services, requests can be sent to the configured compute object request service for compute intensive tasks. Results delivered from the computation are delivered for the compute object response service, giving the desired information back the one of the application components. Compute Envelopes (CEN) can be used for exchange of the compute requests.

The resources column does provide compute resources for processing and computing as well as storage resources for object storage. Commonly these resources are separated for backend use with high end applications customised on the compute resources.

Application components may trigger storage tasks using a storage object request service. Data objects are handled by the service and delivered to the storage resources. Request for retrieval from the storage are handled by the storage object response service. Object Envelopes (OEN) can be used for exchange of the object requests.

For enabling overall scalable integrated systems, mostly for large and voluminous data, the computing and storage resources can communicate for using stored data from within compute tasks and for provisioning and staging of data.

These services are so far using a loosely coupled parallel computing, parallelised on the application component level. Each single task can itself contain scalable and loosely to highly parallel computing jobs running on the available compute resources. MPI and OpenMP can be used here. The CEN Envelopes are used to transfer the tasks and their description.

The user has to ensure that with using the resources the interactivity and latencies of the integrated system still result in appropriate and usable comprehensive system.

Among the compute and storage resources a provisioning and staging mechanism for data and resources requests and responses can be used. Therefore triggering of computing for storage operations and triggering of storage operations for computing are available.

VII. Time Dependence

The same reason why opening large resources for information system purposes is desirable, there is still a dependence on time consumption for interactive and batch processing. Table I shows the characteristic tasks and times that have been considered practical [9] with the current information system applications, for example with environmental monitoring and information system components.

Table I
Tasks and times regarding the overall integrated systems.

Task	*Compute / Storage Times*
In-time events requests	1–3 seconds
Interactive requests	up to 3 minutes
Data processing	1–24 hours
Processing data transfer	n days
Object storage interval	n weeks
Object archive interval	n years

The different tasks afford appropriate policies for interactive and batch use inside the resources network. Besides that, the user and the developer of the application components can use the computing and storage interfaces in order to extend the application facilities using these non-local resources.

Nevertheless for configuring the system and for implementing new operations the decisions have to be made which type of implementation would be more suitable.

Interactive request are mostly not acceptable when response time are longer than a few minutes, even for specialised information systems. HPC systems have shown a good performance for parallelisation of interactive subjobs, being in the range of minutes. Whereas distributed resources are much less scalable and provide less performance due to smaller and mostly different resource architecture types and non-exclusive use. Compute times for 1 to 24 hours will force to decide about the field of operation of the system application, when assigning the tasks and events. For example those compute resources doing computation on large jobs are the computational bottleneck for interactive use. One the other hand for information system purposes, for example needing visual updates within longer intervals, like for special monitoring purposes for environmental, weather, volcano or catastrophes monitoring and using remote sensing, this scenario is very appropriate. Storage resources and object management can reduce the upload and staging times for objects that can be used more than once. Service providers are confronted with the fact that highly performant storage with reliable and long time interval archiving facilities will be needed at a reasonable price.

VIII. Implemented system

The system implemented integrates the component features described from the projects and case studies. Figure 2 shows the implementation of the integrated systems and resources. The components were taken from the GEXI case studies and the well known actmap components [9], [10]. These components handle information like spatial and remote sensing data, can be used for dynamical cartography and trigger events, provide IPC and network communication, and integrate elements from remote compute and storage resources as available with existing compute resources [5], [6], [7]. Processing and computing tasks can for example consist of raytracing, seismic stacking, image transformation and calculation, pattern recognition, database requests, and post processing. The modularisation for development and operation of advanced HPC and application resources and services can improve the multidisciplinary cooperation. The complexity of operation and usage policies is reduced.

A. Application components

The integrated system is built in three main columns, application components in use with scientific tasks for various disciplines, meaning the conventional scientific desktop and information system environment, services, and resources. These columns are well understood from the Grid-GIS house framework. In opposite to the conventional isolated usage scheme, interaction and communication is not restricted to happen inside the disciplines and resources columns only. Non isolated usage can speed up the development of new components and the modification of existing components in complex environments. The workflows with the application scenarios (Figure 2) are:

a) Application communication.
b) Storage task.
c) Compute task.

These tasks can consists of a request, triggered by some event, and a response, when the resources operation is finished. The response can contain data with the status or not, in case that for example an object has been stored on the resources. Based on this algorithm, task definition can be reasonably portable, transparent, extendable, flexible, and scalable.

B. Application communication

A) Request: The internal and framework-external application is triggered from within the framework components (rasmol is used in the example). From within an actmap component a task to an application component is triggered. IPC calls are used with data and information defined for the event.

Response: A framework-external application is started (rasmol locally on the desktop). The external application can further be triggered from the applications available.

C. Storage task

B) Request: From within an actmap component a storage task is triggered. The stored OEN definition is used to transmit the task to the services. The services do the validation, configuration checks, create the data instructions and initiate the execution of the object request and processing for the resources.

Response: The processing output is transmitted to the services for element creation and the element (in this example a photo image) is integrated into the actmap component.

D. Compute task

C) Request: From within an actmap component a compute task is triggered. The stored CEN definition is used to transmit the task to the services. The services do the validation (configuration checks, create the compute instructions and initiate the execution of the compute request and compute job for the resources.

Response: The processing output is transmitted to the services for element creation and the element (in this example a remote sensing image and vector object) is integrated into the actmap component.

IX. Evaluation

The target has been to integrate application communication, computing, and storage resources for handling computing requests and content for distributed storage within one system architecture. The technical details of the components have been discussed in several publications and used in applications publically available. The case study has demonstrated that existing information systems and resources can be easily integrated using envelope interfaces in order to achieve a flexible computing and storage access. As the goal has been to demonstrate the principle and for the modular system components used and due to the previous

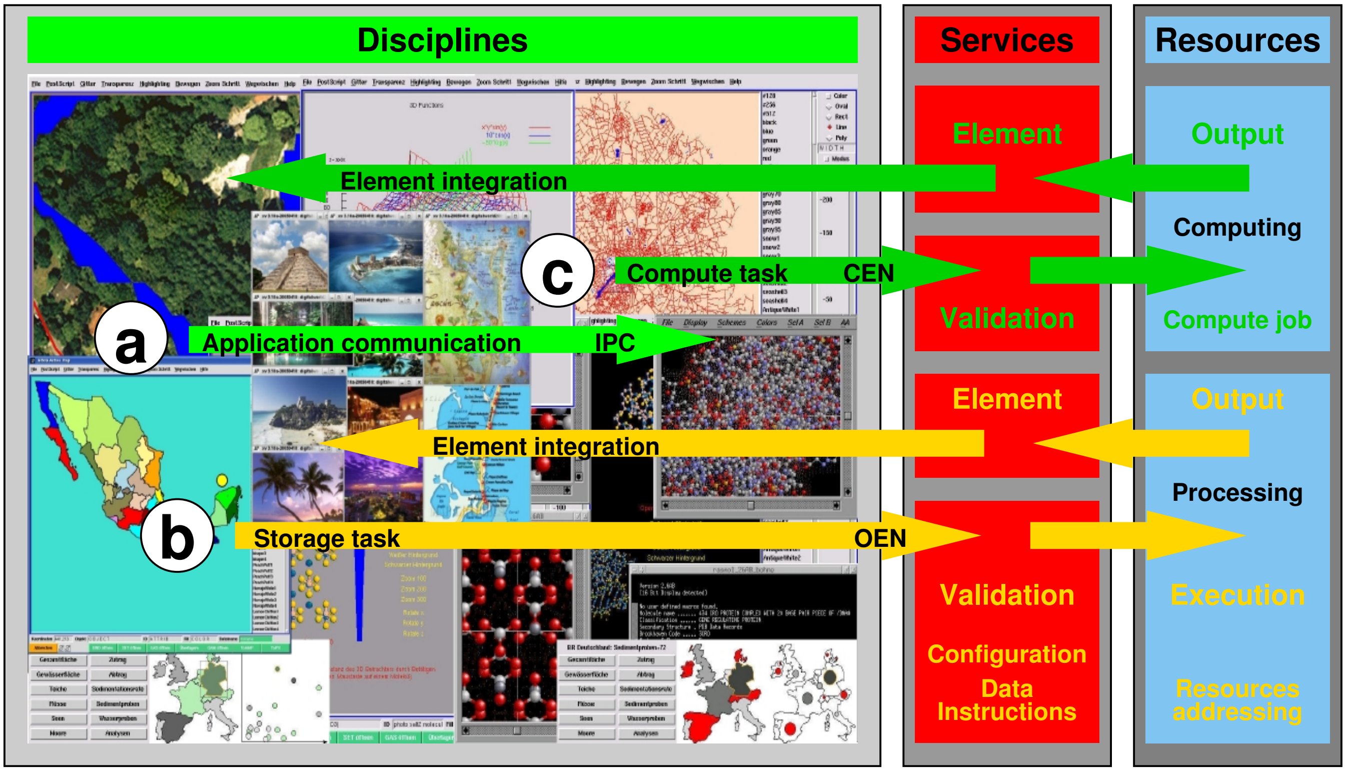


Figure 2. Implementation of the different tasks with integrated systems and resources.

experiences, the services necessary for integration afforded minimal scripting work.

With modern information and computing systems object management is a major challenge for software and hardware infrastructure. Resulting from the case studies with information systems and compute resources, signed objects embedded in OEN can provide a flexible solution.

The primary benefits shown from the case studies of this implementation are:

- Build a defined interface between dedicated information system components and computing system components.
- Uniform algorithm for using environment components.
- Integration of information and computing systems.
- Speed-up the development of new components and the modification of existing components in complex environments.
- Portable, transparent, extendable, flexible, and scalable.
- Hierarchical structured meta data, easily parsable.
- OO-support (object, element) on application level.
- Multi-system support.
- Support for signed object elements, validation and verification via PKI.
- Usable with sources and binaries like Active Source.
- Portable algorithms in between different object and file formats, respecting meta-data for objects.
- Original documents can stay unmodified.
- The solution is most transparent, extendable, flexible, and scalable, for security aspects and modularisation.
- Handling of cooperation and operation policies is less complex [11].
- Guaranteed data integrity and authentication derived from the cryptographic strength of current asymmetric algorithms and digital signature processes.
- Flexible meta data association for any object and data type, including check sums and time stamps.

Main drawbacks are:

- Additional complexity due to additional resources and system environment features like batch scripting (Condor [12], Moab / Torque [13]) and using verification/PKI.
- Complexity of parsing and configuration.
- Additional software clients might come handy to handle resources and generate, store and manage associated data and certificates.

The context is an important aspect, though it cannot be called “drawback” here. With closed products, e.g., when memory requirements are not transparent, it is difficult for users to specify their needs. Anyhow, testing is in many cases not the answer in productive environments. Separate measures have to be taken to otherwise minimise possible problems and ease the use of resources in productive operation.

Even in the face of the drawbacks, for information systems making standardised use of large numbers of accesses via the means of interfaces, the envelopes can provide efficient management and access, as programming interfaces can.

X. Lessons Learned

Integrating information system components and external resources has provided a very flexible and extensible solution for complex application scenarios. OEN and CEN, based on generic envelopes, have provided a very flexible and extensible solution for creating portable, secure objects handling and processing components with integrated information and computing systems.

The case study showed that very different kinds of object data structures and instruction sets may be handled with the envelopes, in embedded or referenced use. Meta data, signatures, check sums, and instruction information can be used and customised in various ways for flexibly implementing information and computing system components. Support for transfer and staging of data in many aspects further depends on system configuration and resources as for example physical bottlenecks cannot be eliminated by any kind of software means.

For future integrated information and computing systems an interface layer between user configuration and system configuration would be very helpful. From system side in the future we need least operation-invasive functioning operating system resources limits, e.g., for memory and a flexible limits management. Homogeneous compute and storage resources and strong standardisation efforts for the implementation could support the use of high end resources regarding economic and efficient operation and use.

XI. Conclusion and future work

It has been demonstrated that integrated information and computing systems can be successfully built, employing a flexible and portable envelopes framework. For this implementation Object Envelopes, Compute Envelopes, and IPC have been used. For the case study Active Source components and Distributed and High Performance Computing resources provided the information system and computing environment. With integrated information and computing systems the following main results have been achieved. Local and inter-application communication can be done using IPC. Object Envelopes can be natively used for handling objects and implementing validation and verification methods for communication. Compute Envelopes can be used in order to define information system computation objects and embed instruction information. These algorithms provide means for generic data processing and flexible information exchange.

The concept used has been found to be least invasive to the information system side as well as to the resources used while being very modular and scalable. The services in between can hold most of the complexity and standardisation issues and even handle products that are meant to be commercially used or licensed. In the future we will have to integrate the features into a global framework for communication purposes and defining standardised interfaces. This implementation has demonstrated a flexible basic approach in order to begin to pave the way and show the next aspects to go on with for future integrated information and computing systems.

Acknowledgements

We are grateful to all national and international academic and industry partners in the GEXI cooperations for the innovative constructive work and to the colleagues at the Leibniz Universität Hannover, IRI, HLRN, WWU, ZIV, D-Grid for participating in fruitful case studies and the participants of the EULISP Programme for prolific discussion of scientific, legal, and technical aspects over the last years.

References

[1] C.-P. Rückemann, “Using Parallel MultiCore and HPC Systems for Dynamical Visualisation,” in *Proceedings, GEOWS 2009, Cancun, Mexico*. IEEE CSP, IEEE Xplore, 2009, pp. 13–18, ISBN: 978-0-7695-3527-2, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4782685&isnumber=4782675 [accessed: 2011-02-20].

[2] C.-P. Rückemann, “Envelope Interfaces for Geoscientific Processing with High Performance Computing and Information Systems,” in *Proceedings, GEOProcessing 2011, Gosier, Guadeloupe, France*. XPS, 2011, pp. 23–28, ISBN: 978-1-61208-003-1, URL: http://www.thinkmind.org/download.php?articleid=geoprocessing_2011_2_10_30030 [accessed: 2011-03-20].

[3] “D-Grid, The German Grid Initiative,” 2008, URL: `http://www.d-grid.de` [accessed: 2009-11-16].

[4] ZIVGrid, “ZIV der WWU Münster – ZIVGrid,” 2008, URL: `http://www.uni-muenster.de/ZIV/Server/ZIVGrid/` [accessed: 2008-12-23].

[5] “ZIVHPC, HPC Computing Resources,” 2011, URL: https://www.uni-muenster.de/ZIV/Technik/ZIVHPC/index.html [accessed: 2011-07-10].

[6] “HLRN, North-German Supercomputing Alliance (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen),” 2011, URL: http://www.hlrn.de [accessed: 2011-07-10].

[7] “ZIVSMP, SMP Computing Resources,” 2011, URL: https://www.uni-muenster.de/ZIV/Technik/ZIVHPC/ZIVSMP.html [accessed: 2011-07-10].

[8] “Open-MPI,” 2011, URL: http://www.open-mpi.org [accessed: 2011-07-12].

[9] “Geo Exploration and Information (GEXI),” 1996, 1999, 2010, 2011, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI (Information) [acc.: 2011-02-20].

[10] “Applications with Active Map Software, Screenshots,” 2005, URL: http://wwwmath.uni-muenster.de/cs/u/ruckema/x/sciframe/en/screenshots.html [accessed: 2011-02-20].

[11] “EULISP Lecture Notes, European Legal Informatics Study Programme, Institute for Legal Informatics, Leibniz Universität Hannover (IRI / LUH),” 2011, URL: http://www.eulisp.de [accessed: 2011-02-20].

[12] “Condor, High Throughput Computing,” 2011, URL: http://www.cs.wisc.edu/condor/ [accessed: 2010-12-26].

[13] “Moab Admin Manual, Moab Users Guide,” 2011, URL: http://www.clusterresources.com/products/mwm/moabdocs/index.shtml [accessed: 2011-02-20].

Open Access Business Model and Financial Issues

Malgorzata Pankowska
Information Systems Department
University of Economics
Katowice, Poland
malgorzata.pankowska@ue.katowice.pl

***Abstract*—The paper covers considerations on business model of open access publication for science and research development. The paper is a "state-of-the-art" report about existing open access repositories. The author aims to present that open access movement strongly depends on financial support of research and science development at academic institutions.**

Keywords-open access; digital repository; business model; financial support

I. INTRODUCTION

The development of the information society and the widespread diffusion of information technology gives rise to new opportunities for research and learning. Higher education institutions have been using Internet and other digital technologies to develop and distribute education and research results for several years. However, much of that materials were locked up behind passwords within proprietary systems, unreachable for outsiders. The trend towards sharing software programmes (open source software) and research outcomes (open access publishing) seems to be strong and complemented by the trend towards sharing learning resources. The reasons for individuals and institutions to use, produce and share open education and science results can be divided into basic technological, economic, social and legal drivers. The technological and economic drivers include improved, less costly and more user-friendly information technology infrastructure, hardware and software. Legal drivers are new licensing schemes that facilitate free sharing and reuse of content. Government-supported educational institutions allow for free sharing and reuse of resources, assuming that open sharing speeds up the development of new learning resources, stimulates internal improvement, innovation and helps the institution to keep good records of materials and their internal and external use.

There is a need to look for new cost recovery models as institutions experience growing competition. Other arguments cover the altruistic motivation of sharing, personal non-monetary gain, such as publicity, reputation and opportunities to reach the market quickly for the competitive advantage. The increase of volume of research work provided online for free is the natural symptom of science development, because storing the knowledge in closed libraries is fated. Knowledge for its further development must be widely distributed, however the credit should be given to all who contributed. The main thesis of the paper is that open access does not mean equal opportunities for participation in science and research development. The paper consists of three parts. The first part includes analysis of open access movement premises, the next subchapter comprises the discussion on financial problems at open repository institutions and an analysis of financial procedures to support the selected open access repositories. The third part includes the business model of open access repositories' development.

II. OPEN SYSTEM DEVELOPMENT PREMISES

Openness in the technical domain is characterized by technical interoperability and functionality. Open standards are important since they make it possible for different software applications to operate together. The openness blurs the traditional distinction between the consumer and the producer. The term prosumer is sometimes used to highlight the blurring of roles. To adapt or modify a digital resource it needs to be published in a format that makes it possible to copy and paste pieces of text, graphics or any published media [1]. Development of Enterprise 2.0, Marketing 2.0 and social media marketing are the excellent examples of prosumers' activities.

On the push side, it is announced that if universities do not support the open sharing of research results and educational materials, traditional academic values will be increasingly marginalized by market forces. On the pull side, a number of possible positive effects from open sharing is put forward, such as: broader and faster dissemination, people involvement in the problem solving, rapid quality improvements and faster technical and scientific development. The free sharing of software, scientific results and educational resources is believed to reinforce societal development and to diminish social inequality [2]. According to Dargan [3], open systems offer a building block approach to development that makes effective use of commercial products and open systems are based on standards that define basic system building blocks and provide a foundation for reuse, interoperability and evolution.

The greatest challenge in designing an open software system is selecting which standards to use for an enterprise. Another challenge is finding suitable standards-compliant commercial products. The third challenge is choosing standards that keep pace with technology innovations.

According to Kavanaugh [4], open source offers, in addition to a very rich set of technologies with long histories, a set of new ways to look at certain problems. Issues include:

- A variety of new licensing options and claims.
- Opportunities to deal with the loosely structured community that creates open source software, from selecting distributions.
- The possibility that open source software is built and maintained in different ways.

Open source software is delivered with source code included or easily available. Generally, intellectual property (IP) covers three main branches - copyright (original artistic and literary works of authorship); patent (inventions of processes, machines, manufactures and compositions of matter that are useful, new and non-obvious) and trademark (commercial symbols) [5]. The copyright and patent acts provide protection for intellectual property against unauthorized use, theft and other violations of the rights granted by those statutes to the IP owner. According to Cronin [6], plagiarism is not a legal but an academic offence which may be punishable according to the institution's regulation. It may be a legal offence if there are intentions to benefit from it financially at the expense of the copyright owner.

Since 1990s the open source software licensing is regulated by the activities of Open Source Initiative. The most important licenses are the General Public Licenses, the Lesser General Public License and the Berkeley Software Distribution (BSD) Licenses. Academics worldwide have started to use open licenses to create a space in the Internet - a creative commons - where people can share and reuse copyright material without fear of being sued. The Creative Commons (CC) license gives others permission to copy, distribute, display and perform the copyright work and derivative works based on it, but for non-commercial purposes only. If anyone wants to use the work for a commercial purpose they must do so in agreement with the right's holders. However, there is no clear understanding of what constitutes commercial use. Another problem is the clause called "Share Alike", meaning that any company trying to exploit the author's work will have to make their added value available for free to anyone else [1]. Researchers, as authors, have plenty of opportunities to support open access and get greater reach for their research - through open-access journals, open-access repositories and author rights management. Hine has noticed that in the science fields where book publishing is the dominant mode of communication and reputation building, publishers have a great deal of control over how those fields are represented and when, how and who can access research outcomes [7].

Though technical standards are necessary for interoperability, there has been a resistance to data standards in many humanities fields because they are perceived as necessitating the standardization of research objects and imposing a normative practice. In human sciences, there is a low-degree of functional dependence and the values and goals incorporated in the technologies of one field are less likely to be shared by another. In physical sciences (e.g., physics, high-energy physics), the knowledge is cumulative, atomistic, concerned with universals, guaranties, simplification, resulting in discovery and explanation. The research works are politically well organized, high publicable and task-oriented. In humanities (e.g., history, linguistics) and pure social sciences (e.g., anthropology, geography), the knowledge is reiterative and concerned with particulars, qualities and resulting in understanding and interpretation. The research works are pluralistic, loosely structured, person-oriented and characterized by low publication rate. In applied sciences (e.g., mechanical engineering), the purposive and pragmatic knowledge is concerned with mastery of physical environment and resulting in products and techniques. The research works are entrepreneurial, dominated by professional values and role-oriented. Patents substitute for publications. In social sciences (e.g., education), the functional and utilitarian knowledge is concerned with enhancement of professional practices and resulting in protocols and procedures. The research works are uncertain in status, dominated by intellectual fashions and power-oriented. Publication rates are reduced by consultancies.

The fields of science have different attitudes towards publishing processes and e-science is differently understood in the particular disciplines. Generally, e-science is defined as the combination of three different developments: the sharing of computational resources, distributed access to massive data sets and the use of digital platforms for collaboration and communication [8]. This is accomplished by transferring the entire research process into the digital environment.

The creation of European collaboration and communication networks in science and scholarly research is one of the key elements of the European Research Area. This means that collaboration and connectivity indicators need to be further developed than they are at present. Scientific collaboration networks i.e. research grids, are constructed to support academic and research community. Some of the notable examples include: TeraGrid, EGEE, LA Grid, and D-Grid [9]. Many different grids have emerged in the last decade. EGEE is used among the European scientific community. BOINC is an open-source software platform for computing using volunteered resources. XtremWeb is an open source software to build lightweight Desktop Grid by gathering the unused resources of desktop computers (CPU, storage, network) [10]. Nowadays, the grid projects provide access to the open repository of research databases and publications, and they create the opportunity to utilize open source software as well as ensure e-publications on projects' deliverables [11]. The open access movement as the worldwide effort was initiated by other organizations to provide free online access to scientific and scholarly research literature, especially peer-reviewed journal articles and their preprints. The open access movement started out with a series of statements and declarations:

- Budapest Open Access Initiative (BOAI),
- Bethesda Statement on Open Access Publishing,
- Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities [12].

Open access means the permission for any user to read, download, copy, distribute, print, search or link to the full texts of the articles, crawl them for indexing, pass them as data to software for processing and use them for any lawful purpose. The number of open access repositories increases. The Registry of Open Access Repositories (ROAR) covers 2172 active tables of repositories. The other worldwide initiative include EBSCO A-to-Z® Web-based tool, DOAJ, Bentham Science, *Open*DOAR. EBSCO A-to-Z® is the industry's most complete Web-based tool for organizing and providing links to all of a library's e-resources, including e-journals, titles in full-text databases, e-journal packages, and e-books. The Directory of Open Access Journal (DOAJ) developed and maintained by Lund University Libraries and the e-Depot of the National Library of the Netherlands (KB) have started a cooperation in order to secure long-term preservation of open access journals. The Swedish Library Association is generously acting as sponsor. The aim of the DOAJ is to increase the visibility and ease of use of open access scientific and scholarly journals thereby promoting their increased usage and impact. Currently, the DOAJ collection covers about 4000 journals and is characterized by a very large number of publishers (2.000+), each publishing a very small number of journals on different platforms, in different formats and in more than 50 different languages. Bentham Science is a major journal publisher of 92 online and print journals, over 200 open access journals and related print and online book series. Bentham Science answers the informational needs of the pharmaceutical, biomedical and medical research community. *Open*DOAR is an authoritative directory of academic open access repositories. Each *Open*DOAR repository has been visited by project staff to check the information that is recorded there. This in-depth approach does not rely on automated analysis and gives a quality-controlled list of repositories. *Open*DOAR aims to provide a quality assured list of academic repositories containing full-text materials that are openly accessible. Open access repositories increasingly play a pivotal role in the emerging research information landscape. Repositories are being deployed in a variety of environments (education, research, science, cultural heritage) and contexts (national, regional, institutional, project, lab, personal). They are operating across administrative and disciplinary boundaries and interact with distributed computational services and social communities. Institutions such as universities, research laboratories, publishers, libraries and commercial organizations are creating innovative repository-linked systems for management of digital content to enable use, reuse and interconnection of information. *Open*DOAR directory includes 19 Polish repositories. Website design for repositories included in the directory is not the same, there is an acceptance of freedom of Website portal design. Although certain general guidelines were considered and approved, the detailed standardization of portal design is demanded in the interest of end user to enable searching and browsing. Nowadays, site map and navigation do not support searching effectively. The end user can recognize the hierarchical construction of repository content, but the construction of sub-repositories and sub-collections within repositories is not so clear and visible, and different names for sub-collections are applied. Lack of standardization of repository design revealed the general tendency to connect the end user with a particular repository even though in the interests of end user is to increase a number of accessible collections.

Taking into account the only one criterion i.e. Ph.D. works access, it should be noticed the only 8 out of 35 Polish digital libraries enable open access to those works. Some other (i.e., 5 universities) ensure open access at local library to printed copies of Ph.D. works. Only the Rector of Polytechnic Institute in Cracow in 2004 has made decision that Ph.D. works are accessible in open repository for all. According to the survey done in 20 other countries the searching results are similar and the general conclusion is that universities do not strongly support the Ph.D. works to be openly accessible online [13]. For Polish digital libraries included in *Open*DOAR directory, the unified standard of metadata, known as Dublin Core version 1.1 is applied for all the stored publications' description. Although interface standard for eLibra digital repository Internet portal was widely applied, information retrieval is not easy because of lack of clear classification of repository content and necessity to browse through a mixture of popular daily news, old manuscripts, maps, and scientific publications.

*Open*DOAR directory does not cover all of scientific research repositories in Poland. Some universities are overlooked and they develop their digital repositories within other projects. For example, Silesian Polytechnic Institute in Gliwice is involved in Springer Open Choice/Open Access scientific publication programme. Within that programme publication are funded in 100% by the Ministry of High Education in Poland within Springer/ICM agreement. Although ICT allows for high speed transfer and mass data storing, it does not mean a permission for uncontrolled redundancy of information. Unfortunately, digital library content classification are not cohesive. Lack of clear classification of publications results in longer time for searching and low effectiveness of information retrieval. Therefore it can be suspected that some valuable research publication are not quoted.

III. OPEN ACCESS REPOSITORY FINANCIAL PROCEDURE

Repositories can be organized as a place to share and exchange resources, which means that people are either users or producers, or they can promote the collaborative production of common resources. Some initiatives of open repositories have institutional backing involving professional staff, others build on communities of practitioners or rely on their voluntary work. The survey of 180 repositories out of 2171 included in the ROAR directory allows for identification of some typical procedures for funding of open repositories.

For example, Norikazu Hyodo from Ochanomizu University Library in Japan reports that the management and operation of the ICT resource rely on the grant from the Japanese government. As of last year, Ochanomizu University Library got a grant from National Institute of Informatics (NII) for open repository development. Similarly, the financial support from NII was provided for the digital library at University of Tokyo. Paul Thirion from University of Liege, Belgium reports that their ORBi - Open Repository and Bibliography is completely financially supported by the library of the university. Ruedi Lindegger from Universität St. Gallen, Switzerland responds that for maintaining the open repository they do not need much money. The applied software is purely open source, but the coordinator for the repository is paid by the research department of the university. The input of the data (publications, projects) is done by the researchers themselves. Some additional improvements are done as a part of the administrative budget of the university with cooperation of two partner institutions.

In Finland, Helda - Digital Repository services are a part of the core functions of Helsinki University Library and thus funded from the library's general operating budget - this includes both the repository management work and the application level technical development and maintenance. For the ICT part (servers, disk space, networking etc.) the repository services use the infrastructure provided by the University IT Department. Director of Bibliothèque de l'EPFL presents that the institutional repository is managed and financially supported partly by the library, partly by the IT service. Jorgen Eriksson perceives that the running, maintenance and development costs of the Lund University repository are part of the yearly budget proposal that the university library applies for from the central university management. So economically it is treated like any other task done by the university library.

HathiTrust Digital Library partners pay the infrastructure costs for the content they deposit. The infrastructure is made up of five elements: storage, data centers, tape backup, servers and miscellaneous hardware, and staff to oversee and maintain these elements. To determine the costs for specific amounts of content, each element is converted into a per GB cost. The per GB costs are then added together to calculate a total per GB per year cost. This total cost includes one storage replacement cycle (e.g., storage that was purchased in year 1 is replaced in year 4 to prevent loss of data). Costs for replacement storage are estimated using 10% reductions in storage costs each year from the time the initial storage was purchased. The total cost also includes the costs of storage and maintenance at two redundant storage locations (one at the University of Michigan and one at Indiana University). Partners are billed on an annual basis and adjustments to the charges are made the following year. HathiTrust is an international community of research libraries consisting of 55 universities.

BioMed Central is a science, technology and medicine research publisher which has pioneered the open access publishing model. Open Repository as a hosted solution from BioMed Central builds and maintains customized digital repositories on behalf of institutions and organizations. Open Repository is a partner of many organizations to support open access and repository development worldwide (e.g., Electronic Information for Libraries, EIFL, which is an international non-profit organization, COAR (Confederation of Open Access Repositories), DuraSpace Registered Service provider that is also a non-profit organization, Symplectic software company, Wijiti as open source software and technology provider).

At the University of Southampton, UK, e-Prints Soton repository is considered to be a core corporate service, alongside HR, student records, finance system and the content management system and is supported as such.

RePEc (Research Papers in Economics) www.repec.org is a collaboration among archive maintainers worldwide who contribute their time to documenting their materials, which are then assembled into a virtual database. RePEc is unfunded. There are various RePEc services, supported by the institutions where they run, but that mainly amounts to provision of hardware. The managers of RePEc services are mainly academics who do not receive compensation nor release time for the work they do to develop and maintain their services. Exceptions may exist for some services, e.g. MPRA at Munich University Library, where those maintaining the service may have that in their job description.

PubMed Central (PMC) as the U.S. National Institutes of Health (NIH) digital archive of biomedical and life sciences journal literature was developed and is operated by the National Center for Biotechnology Information (NCBI), a division of the National Library of Medicine (NLM) at the U.S. National Institutes of Health (NIH). PubMed Central charges publishers nothing for including journal content in the PMC archive. A publisher is responsible for any costs it may incur in creating files that meet PMC's technical standards and transmitting them to PMC. UK PubMed Central is a service of the UKPMC Funders Group working in partnership with the British Library, University of Manchester and the European Bioinformatics Institute in cooperation with the National Center for Biotechnology Information at the US National Library of Medicine (NCBI/NLM) It includes content provided to the PubMed Central International archive by participating publishers.

Publishing Network for Geoscientific & Environmental Data (PANGAEA) as an open access library is hosted by Alfred Wegener Institute for Polar and Marine Research in Bremerhaven and Center for Marine Environmental Sciences (MARUM) at University of Bremen in Germany. The PANGAEA is supported with funding by the European Commission, the Federal Ministry of Education and Research, Deutsche Forschungs gemeinschaft, International Ocean Drilling Program.

Hyper Article en Ligne (HAL) is a multi-disciplinary open access archive for the deposit and dissemination of scientific research papers, whether they are published or not, and for Ph.D. dissertations. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

IV. OPEN ACCESS BUSINESS MODEL

Some years ago, the business model was synonymous with the revenue model. The overall business model now demands specifying the following component models:

- Research results model: how the research institutions create or acquire the knowledge products.
- Distribution model: how the research institutions deliver or distribute the knowledge products to the other researchers.
- Marketing model: the persuasive methods researchers and custodians of knowledge use to promote research results in socio-economic environment.
- Revenue model: how the research institutions get revenue.

The research results, distribution, marketing and revenue model can be combined into one business model, specifically tailored to the particulars of knowledge products, audience, distributions and transactions. The issue of who pays for open access article-processing charges is still being discussed. In Figure 1, the business model covering research results generating and distributing is presented. Generally, research works are funded from the national or regional budgets as well as from private sources and they are conducted by research and development (R&D) units, universities and private companies. Today, the complexity of research process demands individuals to be strongly involved in studying the research results of others, therefore researchers cannot work independently on knowledge production and dissemination activities of others, even if they declare as not affiliated to any institutions. However, they can present the research results on their own Websites. Today, only a small percentage of all articles have been self-archived, but universal online access could be achieved through the open repositories. Research results are published by commercial publishers as well, as fully or partly open access organizations. Open access publishers are sponsored by individuals (pay-for-publish approach) or institutions (donation approach), therefore the open repositories can offer publications free of charge to the readers. Simultaneously, publications' market is supplied by commercial publishers, who sell books and journals to book stores and authorized access libraries. The commercial publishers prefer pay-for-access model as well as pay-for-publish. In the latter, researchers spend money from research grants on publications.

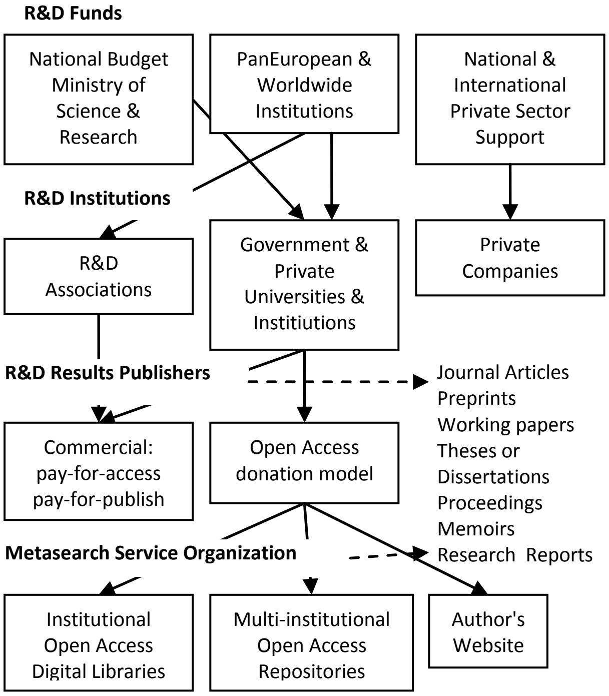

R&D Funds
National Budget Ministry of Science & Research
PanEuropean & Worldwide Institutions
National & International Private Sector Support
R&D Institutions
R&D Associations
Government & Private Universities & Instituitions
Private Companies
R&D Results Publishers
Journal Articles
Preprints
Working papers
Theses or Dissertations
Proceedings
Memoirs
Research Reports
Commercial: pay-for-access pay-for-publish
Open Access donation model
Metasearch Service Organization
Institutional Open Access Digital Libraries
Multi-institutional Open Access Repositories
Author's Website

Figure 1. Open access repository financial support.

Although the commercial models are well applied in developed countries, the open access movement encourages the researchers having different research and academic backgrounds from all around the globe to submit their contributions online. The researchers understand that open access is an effective way to reach the appropriate audience. They know that when managing the intellectual property, they are able to maximize the value of the intellectual property, not to maximize the protection of closed resources of knowledge. Commercial publishing house demands authors to be involved in the process of books' and journals' editing. In that way they can reduce the intellectual property dissemination costs. Generally, the open access repositories are financially supported by the university and regional libraries which receive special funds from government institutions (i.e. the Ministry of Science, Research and Education), as well as from international projects as it is visible in the presented above review. However, strong support by the non-profit organization and private companies is still required. The open access movement and open repositories are a way to reduce costs of intellectual property production and dissemination, because here authors are also requested to work on editing and to be involved in peer reviewing. The commercial publisher implements blended model of distributing books and journals traditionally in printed versions as well as online. However, in the knowledge generating process the costs of production are still increased, therefore in the research results publication process, the budget is spent mostly on research and development process.

V. Conclusion

The users need an improvement of access to open resources. The rapidly growing number of repositories makes it important to find the most relevant and highest quality resources. Metadata may improve the function of search engines, therefore approaches such as automatically generated metadata and folksonomies are being tested as applicable. There is an imbalance between the provision of open repositories and their utilization. Taking into account 2172 repositories included in the Registry of Open Access Repositories (ROAR) it should be noticed that vast majority of open repositories is not based on Western European culture, but they are developed in Japan, China and in Latin America countries. A number of projects exists in countries to support open repositories based on their own language and cultures. There is a risk, because this way the researchers share their knowledge among one culture country, taking into account that it can be a small and less developed country. Although there is an idea of repurposing the materials and the interoperability is a key issue for further open access movement development, the worldwide knowledge exchange is not simple, easy or even rational task. The digitalization of knowledge results will be helpful but does not solve the problem of understanding research results. Open standards implementation is necessary to enable research resources searching across repositories and downloading, integrating and adapting across platforms. Therefore, the development of open standards is a specialized task, which requires financial support. Considering the presented above financial procedures, there is a general conclusion that the vast majority of repositories is funded by national and international organizations, therefore there is a risk of consigning less developed countries to playing the role of consumers. The most frequented quoted publications are the results of research done in the well developed countries.

Acknowledgment

Hereby, I would like to thank all managers of open repositories, who supported me in survey on financial procedures.

References

[1] "Giving Knowledge for Free, The emergence of open education resources," Centre for educational research and innovation, Organisation for economic co-operation and development, OECD, 2007.

[2] A. Swan, "Open Access and the Progress of Science" American Scientist, May-June 2007, pp. 197-199.

[3] P.A. Dargan, "Open systems and standards for software product development," Artech House, Boston, 2006.

[4] P. Kavanaugh, "Open Source Software, Inplementation and Management," Elsevier, Digital Press, Amsterdam, 2004.

[5] B.T. Yeh, "Intellectual Property Rights violations: Federal Civil remedies and criminal penalties related to Copyrights, Trademarks and Patents CRS Report for Congress,"2008, http://www.fas.org/sgp/crs/misc/RL34109.pdf, access date: 4 January 2010.

[6] C. Cronin, "Plagiarism, copyright, Academia and Commerce," presentation given at Colby College in Proceedings of the Conference on Information Ethics and Academic Honesty, 2003, http://abacus.bates. edu/cbb/ docs/ Cronin.pdf , access date: 4 January 2010.

[7] Ch.M. Hine, "New Infrastructures for Knowledge Production, Understanding e-Science," Information Science Publishing, Hershey, 2006.

[8] T. Hey, and A.E. Trefethen, "The UK e-Science Core Programme and the Grid," Future Generation Computer Systems, 2002, 18(8), pp. 1017-1031.

[9] A.B. Mohammed, J. Altmann, and J. Hwang, "Cloud Computing Value Chains: Understanding Businesses and Value Creation in the Cloud," Economic Models and Algorithms for Distributed Systems, Birkhauser Verlag Basel, 2009, pp. 187-208.

[10] S. Naqvi, M. Villari, J. Latanicki, and P. Massonet, "From Grids to Clouds - Shift in Security Services Architectures," in Cracow '09 Grid Workshop Proceedings, Bubak M. Turala M., Wiatr K., Eds. Cracow, Poland, October 12-14, 2009, pp. 28-36.

[11] "Tera Grid. Science Gateway Home," 2011, http://www.tergrid.org/web/science-gateways/. access date: 18 July 2011.

[12] "Budapest Open Access Initiative: Frequently Asked Questions," August 4, 2010 http://www.earlham.edu/ ~peters/fos/boaifaq.htm#openaccess. access date: 10 January 2011.

[13] M. Pankowska, "Open Access Movement in Science and Research", International Conference on Information Society, iSociety'2011, Shoniregun Ch.A. Akmayeva G.A. (eds.) Infonomic Society, London, pp. 347-353.

Websites List

[14] Bibliothèque de l'EPFL, http://library.epfl.ch

[15] BOINC, http:// boinc.berkeley.edu/

[16] DOAJ, http://www.doaj.org/

[17] D-Grid, http://www.d-grid.de/

[18] EBSCO A-to-Z$^{®}$ http://atoz.ebsco.com/

[19] EGEE, http://public.eu-egee.org

[20] e-Prints Soton, http://eprints.soton.ac.uk/

[21] HAL, http://hal.archives-ouvertes.fr/

[22] HathiTrust Digital Library, http://www.hathitrust. org/cost

[23] Helda - Digital Repository, https://helda. helsinki.fi/

[24] LA Grid, http://latinamericangrid.org

[25] NII, http://www.nii.ac.jp/en/

[26] *Open*DOAR, http://www.opendoar.org/

[27] Open Source Initiative, http://www.opensource.org

[28] Open Repository, http://www. openrepository.com/

[29] PANGAEA, http://www.pangaea.de

[30] PMC, http://www.ncbi.nlm. nih.gov/pmc/

[31] ROAR, http://roar.epronts.org/

[32] TeraGrid, http://www.teragrid.org

[33] XtremWeb, http://www. xtremweb. net/

Study of the Growth of a New Social Network Platform

Ning Tang
Mobile Life and New Media Laboratory
Beijing University of Posts and Telecommunications
Beijing, China
ning.tang0@gmail.com

Chunhong Zhang
Mobile Life and New Media Laboratory
Beijing University of Posts and Telecommunications
Beijing, China
zhangch.bupt.001@gmail.com

***Abstract*—Nowadays, with the rapid development of social network sites (SNS), many efforts have been made on the analysis of such networks, which is important for us to enrich our social network knowledge and improve the SNS design. However, most analysis have focused on the very popular SNS, there is little comprehensive analysis of growing process of a SNS from the exact original stage, therefore, lacking of deep understanding how the SNS evolve over time. In this paper, a comprehensive growth data set of a new social network site constructed by our lab is examined to see the growing process of the network and to find some underlying growth mechanisms. This observation provides an empirical evidence of the Barabási-Albert (BA) model. During the observation, the number of new added links is shown to have a linear relationship with the number of new added users and there is a preferential attachment phenomenon in the link formation process. In addition, the degree distribution at different time points is presented to see the formation of the power-law distribution. Finally, the topological properties are found to be stable after a period of development, though the network is still growing.**

Keywords-social network sites; growth; power-law degree distribution; topological properties.

I. Introduction

There are various social network sites nowadays. On SNS, users can make friends, share pictures, share videos, write blogs and play games. Despite the different goals and purposes of various social network sites, they have been shown to have a number of common structural features, such as power-law degree distribution, small diameter, and high clustering coefficient [1][2]. Many previous analysis have focused on validating static common features in different popular social network sites, however, there is little comprehensive analysis of the growing process of these features from the exact original stage of a SNS, therefore, lacking of deep understanding how the features form over time. Although there are some theoretical models [3] that try to reveal the underlying mechanisms, an empirical view of the formation process is desired.

An in-depth understanding of the growing process of a SNS from the exact original stage can help us learn the underlying mechanisms which result in the specific features. Mastering the mechanisms enables us to simulate similar social networks and provides a possibility for further social networking study. What is more, a better knowledge of the evolution of the topological properties allows us to predict the future of networks and make corresponding improvements.

In this paper, an empirical analysis of the growing process of a new social network site from the exact original stage is presented to see how its features shape over time. The social network is constructed by our lab. The dataset used in this paper is the testing data extracted from the server for 27 consecutive days. Compared with the crawled data sets generally used in previous studies [2][4][5][6], the data set used in this paper is a complete and time continuous one which may help us make a more comprehensive and more accurate understanding of the whole network.

The number of new added links is found to have a linear relationship with the number of new added users—every 11 additional edges will bring about 6 new users. And there is a preferential attachment phenomenon in the link formation process—more than 55% of new links are attached to the top 30% large-degree nodes. Next, the degree distribution at different time points is plotted to see the dynamic shaping of the power-law distribution. Finally, the topological properties are found to be stable after a period of development, though the network is still growing.

The rest of this paper is organized as follows: Additional background and related works are provided in Section 2. In Section 3, the methodology for obtaining the data set and its limitations are described. In Section 4, an empirical analysis of the growing process of the new social network is presented. Finally, the conclusion remarks and future work are drawn in Section 5.

II. Background and Related Work

A. Topological Properties

There are some general metrics to characterize the SNS, which are called topological properties.

- Degree. Degree is the number of edges incident to a vertex of a graph. The node degree distributions of many large-scale social networks have been shown to conform to power-laws. Power-law networks, also known as scale-free networks, are networks where the probability that a node has degree k is proportional to $k^{-\gamma}$, for large k and $\gamma > 1$. The parameter γ, whose value is typically in the range $2 < \gamma < 3$, is called the power-law coefficient. In power-law network, the majority of the nodes have small degrees, but a few nodes called hubs have significantly high degrees. In 1998, Barabási and Albert found that growth and preferential attachment were two important mechanisms of the formation of power-law networks [3]. Growth means that the number of nodes in the network increases over time. Preferential attachment means that new links are more likely to attach to large-degree nodes. They later proposed the Barabási and Albert (BA) model based on the two mechanisms.

- Diameter. The shortest path length L between two vertices in a graph is the number of edges in a shortest path connecting them. The average shortest path length <L> is the mean of L between any pairs that have at least a path connecting them. Diameter D is defined as the maximum of the shortest path length.
- Clustering coefficient. Clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups with a relatively high density of ties [7][8]. Clustering coefficient of node i in undirected network is defined as

$$C_i = 2E/ K (K—1). \tag{1}$$

That is the ratio of the number E of edges that actually exit between the k neighbors of node i to the potential number k (k—1)/2. The clustering coefficient C of the whole network is the average of all individual C_i .

Studies have shown that the Web [9][10], scientific collaboration on research papers [11], film actors [12], and general social networks [13] have small-world properties. Small-world networks have a small diameter and exhibit high clustering.

B. Related Works

Recently, much work has focused on understanding the structure and evolution of large-scale online social networks. [2][4], present empirical analysis of statistical properties and validate the power-law, small-world and scale-free properties of several large-scale social networks. Bimal [5] finds that while the individual links that constitute the activity network change rapidly over time, the average network properties remained relatively stable in Facebook. Alan [6] shows that new link formation in Flickr follows preferential attachment, but the link creation process cannot be explained by the BA model alone because users are far more likely to link to nearby users than that model would suggest.

Above researches are all about popular social networks which have already processed the common features. But the tracing observation of the formation of a new SNS, which is helpful for a better understanding of the underlying growth mechanisms and helpful for predicting the future of networks, is lack of studying.

III. METHODOLOGY

To get the growth data of a SNS from the exact original stage, a new SNS is constructed by our lab. Then the new SNS began to get a test from September 15th, 2010. The dataset used in this paper is the user data extracted from the server at the end of the test on October 11th. During the 27 days, 140 users have registered on the platform, including 110 undergraduates in four classes at Beijing University of Posts and Telecommunications and 30 graduate students in Mobile Life and New Media Laboratory. The dataset contains all the link formation information of the 27 days, including the creator, the target and the timestamp.

In contrast to the crawled data sets used in other papers [2][4][5][6], the data set used in this paper is a complete and time continuous one, which may help us make a more comprehensive and accurate understanding of the whole network. However, although our dataset contains all the user data, the statistical properties may not be so obvious because the number of testing users is small and the testing period is short.

IV. DATA ANALYSIS

The network is composed of users (vertices) and links (edges) among them. Since link creation in our network requires consent from the link target, a link connecting the creator and the target is undirected. Among the 140 registered users, there are 47 users who have at least one link with others, while the remaining 93 people are isolated nodes. In the next analysis, we'll only consider users that connect with others.

In order to learn the evolution of the network, the growing process of the network is divided into segments. During the test, registration is mainly concentrated in a few days and the number of registers varies greatly every day (from 0 to 49 per day). To make the partition as even as possible, the growing process is divided by the number of new added links instead of by days. Finally, the network developed to have 77 edges. Considering granularity, the growing process is finally divided into seven segments. In each segment, 11 edges are added into the network. In the following analysis, the network is examined at each end of the seven segments. Table Ⅰ shows the number of vertices and edges at the seven time points.

TABLE I. THE NUMBER OF VERTICES AND EDGES AT SEVEN TIME POINTS

Time point	1	2	3	4	5	6	7
Edges	11	22	33	44	55	66	77
Vertices	10	15	22	30	36	41	47

ISBN: 978-1-61208-161-8

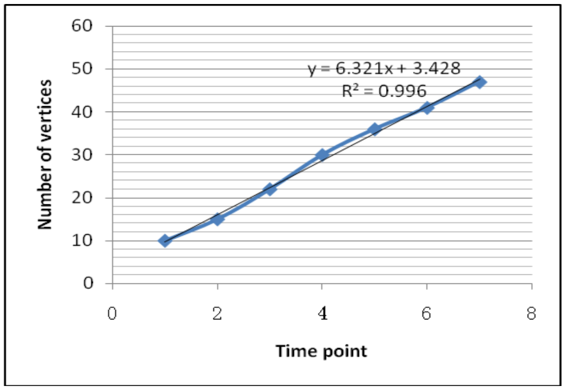


Figure 1. Plot of the number of vertices at seven time points

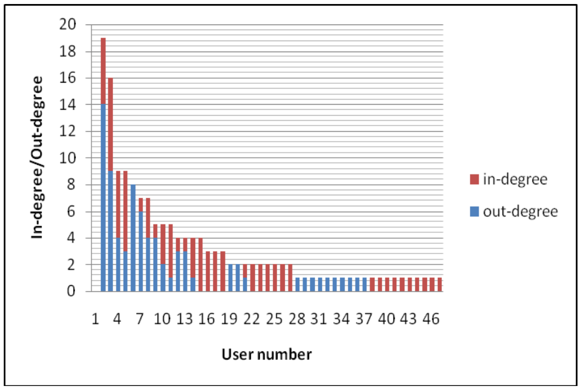


Figure 2. The statistics of the in-degree and out-degree of each user

A. *Preferential Attachment Phenomenon*

The well-known Barabási-Albert (BA) model has been shown to result in networks with power-law degree distributions. In BA model, new links are attached to nodes using a probability distribution weighted by node degree. Since the dataset used in this paper is very small, there could not be a statistic of the link formation distribution with degree. Instead, the number of links that are attached to nodes whose degrees rank in the top 30% at the former time point is calculated to examine whether there is a preferential attachment phenomenon in the link formation process. The result is shown in Table Ⅱ.

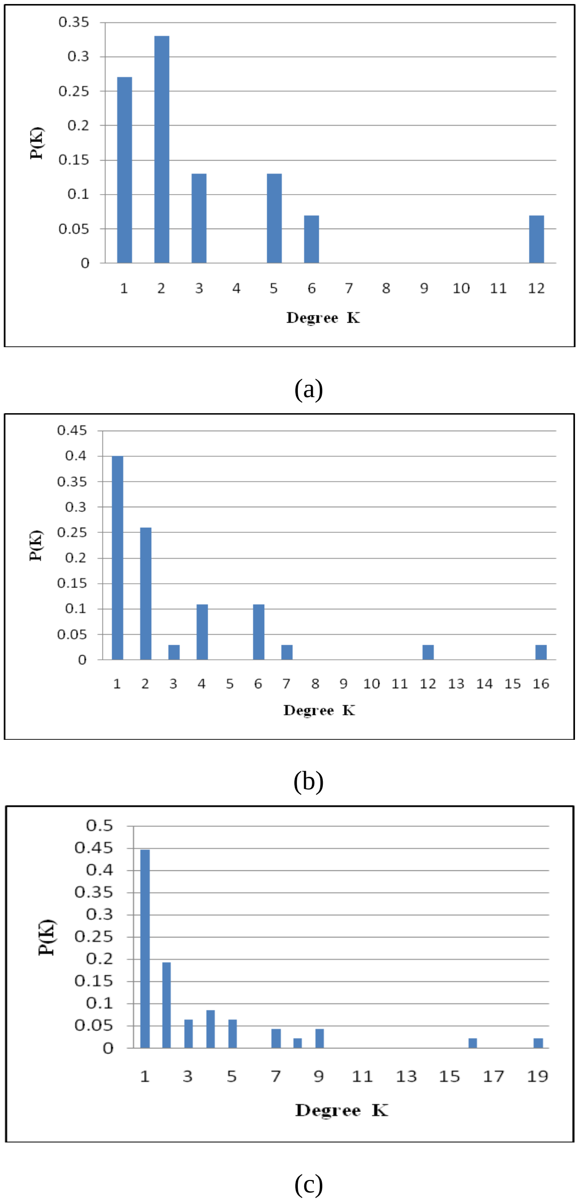
It is obvious that there is a preferential attachment phenomenon—more than 55% (6/11) of new links are attached to the top 30% large-degree nodes.

Since the establishment of a link needs a creator and a target, so the link can be regarded as a directed one in this respect. Here we define the out-degree of a node as the number of links that it creates, and the in-degree as the number of links that it receives. Figure 2 is the statistics of the in-degree and out-degree of each user, ranking in descending order of the total degree. It is obvious that large-degree nodes are more likely to be creators.

From Table Ⅱ and Figure 2, the underlying mechanism which results in the power-law degree distribution can be figured out. The large-degree nodes are generally active users who like to create links, so new links are more likely to be established between large-degree nodes and new added vertices. What is more, since every 11 additional edges generally bring about 6 new users, new nodes are continuously added to the network, making the degree distribution more and more asymmetrical, as shown in Figure 3—more and more nodes have small degrees, while a few hub nodes have larger and larger degrees.

TABLE II. THE NUMBER OF LINKS THAT ARE ATTACHED TO THE TOP 30% LARGE-DEGREE NODES

Time point	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Edges	-	7	8	6	9	6	6



(a)

(b)

(c)

Figure 3. The change of degree distribution at three time points.

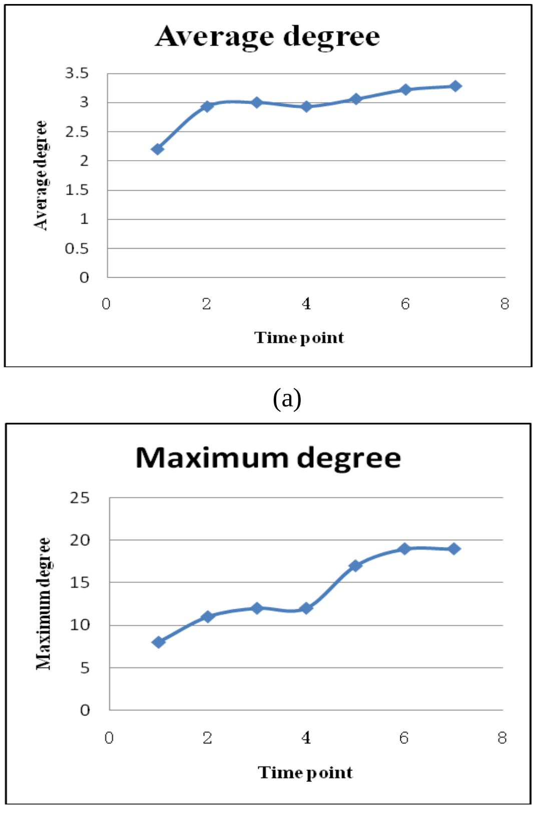
ISBN: 978-1-61208-161-8

B. Topological Properties

Next there will be a look at the evolution of the topological properties. Table Ⅲ is the detailed growing information at seven time points. Figure 4 plots the changing processes of these properties. It is shown that although the network keeps a same expanding speed during each time point, all the properties tend to be stable from the fifth time point. The network seems to have entered a stable stage.

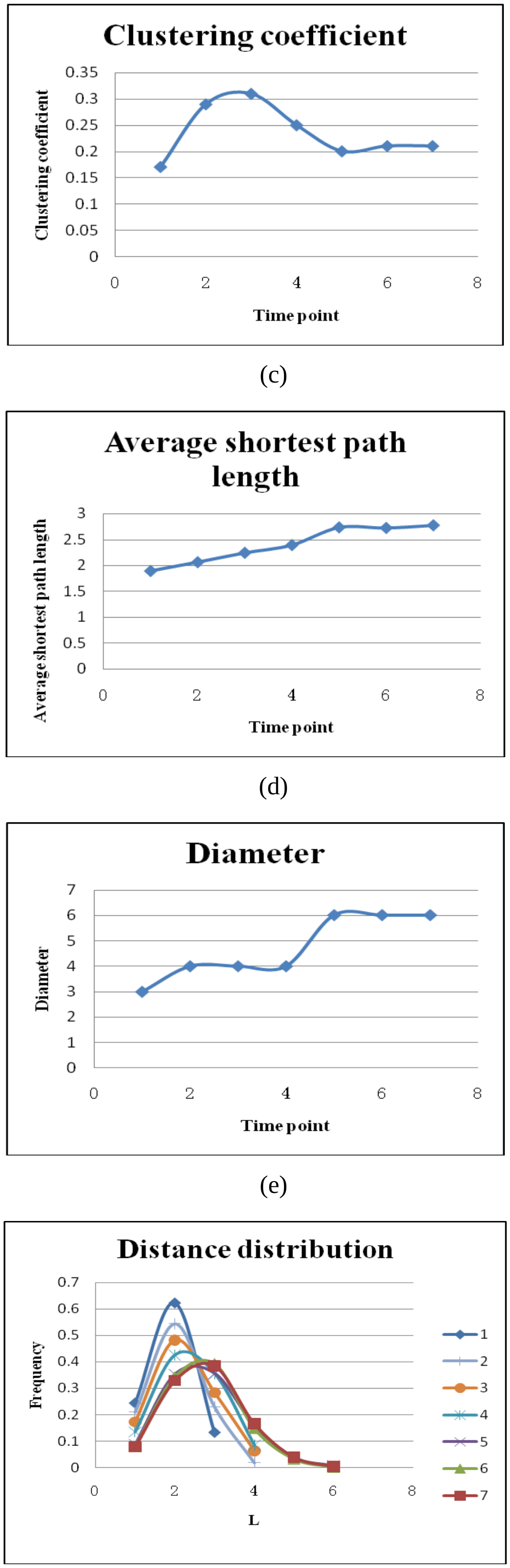
TABLE III. THE TOPOLOGICAL PROPERTIES AT SEVEN TIME POINTS

Time point	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Average degree	2.2	2.93	3	2.93	3.06	3.22	3.28
Maximum degree	8	11	12	12	17	19	19
Clustering coefficient	0.17	0.29	0.31	0.25	0.2	0.21	0.21
Average shortest path length	1.89	2.06	2.24	2.39	2.73	2.72	2.77
Diameter	3	4	4	4	6	6	6



(a)

(b)



(c)

(d)

(e)

(f)

Figure 4. The evolution of the topological properties

In order to affirm the new developed network used in this paper has the common features that previously observed in other large-scale social networks and ensure the above study is applicable to general social networks, there is a check of the power-law and small-scale properties in the new network. Figure 5 is the degree distribution of the final network.

In Figure 5, majority nodes have small degrees (k=1, 2), some nodes have moderate degrees (k=3~9), few nodes (the long tail) have very large degrees (k=16, 19). It presents an approximate power-law distribution. To a power-law network, the cumulative degree distribution also follows a power-law form as $P(\geq K) \propto k^{-(\gamma-1)}$. The above function plotted in log-log coordinates is a straight line with slope －(γ—1). Figure 6 is the log-log plot of the cumulative degree distribution. Using linear least squares regression to get the slope －(γ—1) = －1.268. So the degree distribution follows a power-law form with γ=2.268 and the network is a scale-free network.

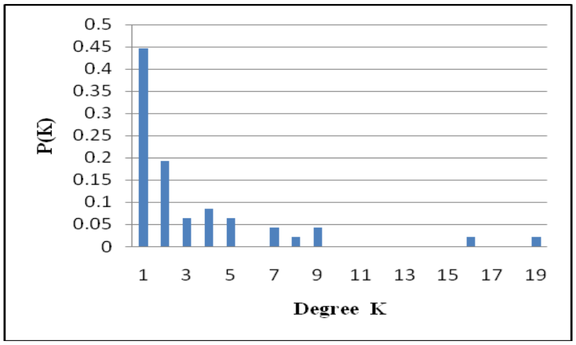


Figure 5. Degree distribution

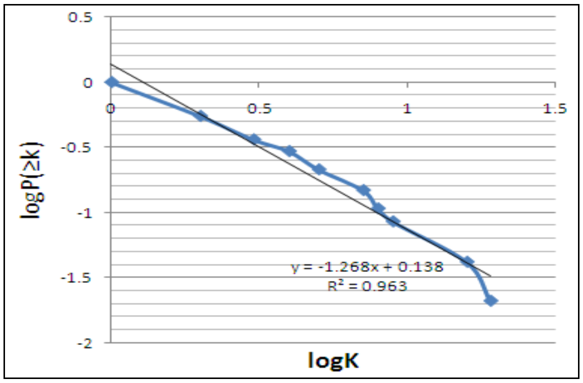


Figure 6. Log-log plot of the cumulative degree distribution

The clustering coefficient C of the whole network is 0.21, much higher than that of a corresponding random graph of the same size C_{rand}=<k>/N=3.28/47=0.07, <k> is the average degree of the undirected network. In addition, the network also has a small average shortest path length 2.77. Hence, small-word property also exists in this network.

At the end of the test, the social network has exhibited the small-world and scale-free properties after a short-term development. Therefore, social networks are born to have the common features. It also indicates that the above study of the new social network in this paper is applicable to general social networks.

V. Conclusion and Future Work

In conclusion, the experiment in this paper provides an empirical evidence of the BA model. Both growth and preferential attachment mechanisms are observed in our observation. In addition, a new phenomenon that the number of new added users has a linear relationship with the number of new added links is observed. The topological properties are found to be stable after a period of development. Finally, the common small-world and scale-free properties are proved to exist in this small-scale social network, indicating that the study of the network is applicable to general social networks.

We think our work provides an insight into the understanding of the original stage of a social network, which will help us know the underlying mechanisms and predict the future growth. Much work remains to be done. We'll make a longer and larger-scale observation of the growing process of the network to get more statistical properties. In addition, other aspects of the network, such as user behaviors, are worth studying.

Acknowledgment

This work is supported by China-Finland Cooperation Project on the Development and Demonstration of Intelligent Design Platform Driven by Living Lab Methodology (2010DFA12780), Architecture and key Technology Research on Broadband Wireless Campus Innovation Testbed (2010ZX03005-003), the National Key Program of New Generation of Broadband Wireless Mobile Communication Network (2011ZX03005-004-02) and Key Laboratory of Universal Wireless Communications, Ministry of Education.

References

[1] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of Topological Characteristics of Huge Online Social Networking Services," In Proceedings of the 16th World Wide Web Conference (WWW'07), Banff, Canada, 2007, pp. 835–844.

[2] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," In Proceedings of the 5th ACM/USENIX Internet Measurement Conference (IMC'07), San Diego, CA, 2007.

[3] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," Science, vol. 286, 1999, pp. 509–512.

[4] Feng Fu, Lianghuan Liu, and Long Wang, "Empirical analysis of online social networks in the age of Web 2.0," Physica, vol. 387, 2008, pp. 675–684.

[5] Bimal Viswanath, Alan Mislove, and Meeyoung Cha, "On the Evolution of User Interaction in Facebook," WOSN'09, August 17, 2009.

[6] Alan Mislove, Hema Swetha, and Koppula Krishna P . Gummadi, "Growth of the Flickr Social Network," WOSN'08, August 18, 2008.

[7] P. W. Holland and S. Leinhardt, “Transitivity in structural models of small groups,” Comparative Group Studies, vol. 2, 1998, pp. 107–124.

[8] D. J. Watts and Steven Strogatz, “Collective dynamics of 'small-world' networks,” Nature, vol. 393 (6684), June 1998, pp. 440–442.

[9] R. Albert, H. Jeong, and A.-L. Barabási, “The Diameter of the World Wide Web,” Nature, vol. 401, 1999, p. 130.

[10] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph Structure in the Web: Experiments and Models,” In Proceedings of the 9th International World Wide Web Conference (WWW’00), Amsterdam, May 2000.

[11] M. E. J. Newman, “The structure of scientific collaboration networks,” Proceedings of the National Academy of Sciences (PNAS),vol. 98, 2001, pp. 409–415.

[12] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, “Classes of small-world networks,” Proceedings of the National Academy of Sciences (PNAS), vol. 97, 2000, pp. 11149–11152.

[13] L. A. Adamic, O. Buyukkokten, and E. Adar, “A social network caught in the Web,” First Monday, 8(6), 2003.

The Simple Genetic Algorithm Performance: A Comparative Study on the Operators Combination

Delmar Broglio Carvalho,
João Carlos N. Bittencourt
Tecnology Department
State University of Feira de Santana (UEFS)
Feira de Santana, Brazil
E-mail: {carvalho.db, joaocarlos}@ecomp.uefs.br

Thiago D'Martin Maia
Department of Mathematical Sciences
State University of Feira de Santana (UEFS)
Feira de Santana, Brazil
E-mail: tdmaia@uefs.br

***Abstract*—This paper presents a comparative and experimental study about the performance of the Simple Genetic Algorithm (SGA) using five classic benchmarking functions. The performance analysis is accomplished on the combination of the operators of reproduction and crossover with the control parameters having been fixed. The overall behavior of the SGA is evaluated by the fitness of the best individual analyzed during the evolution and at the ending of the same one. The results that are presented show that the SGA can be effective and competitive to optimization on a test suite of benchmark functions.**

***Keywords*-Genetic Algorithm; Parameterization of GA; Generational Replacement Model; Single-Point Crossover; Uniform Crossover.**

I. Introduction

The Simple Genetic Algorithm (SGA) presented by Goldberg [1] plays an important role in the use of the approaches based on the dynamics of natural genetics and still being a study target [2]. This proposal has been used in the implementation of many derived approaches, and many researchers have drawn the performance analysis of the Genetic Algorithms (GAs) basing its studies in the control parameters (populations size, crossover and mutation rates) and their potential adjustment [3]. Normally, this parameterization depends on the knowledge of the designer about the problem definition; of the values attributed to the parameters and of the adequate choice of the used methods to implement the operators. In this universe of choices, each designer can create a particular algorithm to a specific problem [4]–[7], being always a generic SGA as the worse one of the implementations.

In this paper, the SGA performance is examined into five benchmarking functions, considering a fixed size of the population and constant crossover and mutation rates. The performance test is carried out through for the combination of three strategies of reproduction and two kinds of crossover. No additional strategy was established, and the benchmarking functions were normalized to facilitate the comparisons.

The present paper is structured as follows: Section II presents the benchmarking functions and Section III describes the methodology used. In Section IV, results of experiments are reported. In Section V, some general conclusions are mentioned.

II. Benchmarking Functions

Many benchmarking functions have been used to perform a stress test of various GA approaches. Digalakis [3] summarizes this set of benchmarking functions, which comes the set of characteristics required for benchmarking tests using GAs. In this set, five functions had been selected to perform the proposed study, which are listed below:

1) F1 function (Sphere): paraboloid function, smooth, unimodal, convex, symmetric, and whose convergence to the global optimum is easily achieved.

$$f_1(x) = \sum_{i=1}^{2} x_i^2 \tag{1}$$

$$-5.12 \leq x_i \leq 5.12$$

2) F2 function (Rosenbrock): considered of high difficulty level resembling a saddle function, imposing strong restrictions on the algorithms that are not suitable to search for directions.

$$f_2(x) = 100\left(x_1^2 - x_2\right)^2 + (1 - x_1)^2 \tag{2}$$

$$-2.048 \leq x_i \leq 2.048$$

3) F3 function (Step): function at representative levels of flat surfaces, which are obstacles for optimization algorithms, whereas the surfaces in the levels do not provide any information about which direction is favorable to the search.

$$f_3(x) = \sum_{i=1}^{5} integer\left(x_i\right) \tag{3}$$

$$-5.12 \leq x_i \leq 5.12$$

4) F4 function (Rastrigin's function): this function represents a surface performance of extreme complexity in the

search for global optimal solution, given the existence of numerous local solutions.

$$f_4(x) = 10 \cdot n + \sum_{i=1}^{2} \left(x_i^2 - 10 \cdot \cos(2\pi \cdot x_i)\right) \qquad (4)$$

$$-5.12 \leq x_i \leq 5.12$$

$$n = dim(x_i)$$

where n represents the numerical value of x_i dimension.

5) F5 function (Foxholes): the main feature of this function is to produce local solutions in an independent environment with a high level of discontinuity.

$$f_5(x) = \left(0.002 + \sum_{j=1}^{25} \left(j + \sum_{i=1}^{2} (x_i - a_{ij})^6\right)^{-1}\right)^{-1} \qquad (5)$$

$$-65.536 \leq x_i \leq 65.536$$

$$(a_{ij})_{2\times 25} = A$$

In this function, the matrix $A_{2\times 25}$ is formed by constants and, in order to simplify the exibition, their values are grouped as follow:

$$C_0 = \begin{bmatrix}-32 & 16 & 0 & 16 & 32\end{bmatrix}$$
$$C_1 = \begin{bmatrix}-32 & -32 & -32 & -32 & -32\end{bmatrix}$$
$$C_2 = \begin{bmatrix}-16 & -16 & -16 & -16 & -16\end{bmatrix}$$
$$C_3 = \begin{bmatrix}0 & 0 & 0 & 0 & 0\end{bmatrix}$$
$$C_4 = \begin{bmatrix}32 & 32 & 32 & 32 & 32\end{bmatrix}$$
$$C_5 = \begin{bmatrix}16 & 16 & 16 & 16 & 16\end{bmatrix}$$

The groups C0-C5 can now be associated to A:

$$A = \begin{bmatrix}C_0 & C_0 & C_0 & C_0 & C_0 \\ C_1 & C_2 & C_3 & C_4 & C_5\end{bmatrix}$$

This subset contains important characteristics of many objective functions found in optimization problems, such as: smoothness, unimodality, multimodality, a very narrowness ridge, a flat surface, and too many local optima [3].

III. Methodology

The objective of this paper is to present an analysis of the SGA performance, in its classic form [1], and to evaluate such performance using a combination of basic methods for reproduction and crossover. The populations size, crossover and mutation probability are used with constant values associated, to avoid the effect of these parameters in the overall analysis.

The stop condition of the evolution was established as a finite number of generations. The population replacement scheme adopted was the Generational Replacement Model (GRM), which replaces the entire population, in each generation, by its offspring. To guarantee the maintenance of the best solution gotten in each previous generation, the Elitist strategy is applied in each next generation [8].

The performance measure may be taken in two moments of the evolution: the first one can be since the initial steps, and is called ongoing analysis, and the other one at the evolution's end, called stopped analysis. The ongoing analysis gives an idea of the evolution until the present generation, and the stopped analysis supplies the best solution found until then. These criteria have been detailed in [1], [3]. In this work, the ongoing analysis was established having the measure being obtained at the 10th generation, and the stopped analysis at the 80th generation.

The selected methods of reproduction are:

- R_1 - Stochastic sampling with replacement (Roulette wheel selection);
- R_2 - Remainder stochastic sampling with replacement;
- R_3 - Stochastic tournament.

The selected crossover methods are:

- X_1 - Single-point crossover;
- X_2 - Uniform crossover.

The reproduction and crossover methods, above listed, were combined to assemble the set of tests. Such set was assigned as follows:

C_{11} - Reproduction method R_1 with crossover X_1;
C_{12} - Reproduction method R_1 with crossover X_2;
C_{21} - Reproduction method R_2 with crossover X_1;
C_{22} - Reproduction method R_2 with crossover X_2;
C_{31} - Reproduction method R_3 with crossover X_1;
C_{32} - Reproduction method R_3 with crossover X_2.

For each item in the set, 100 independent runs of the SGA were carried out. For each run, the best individual's evolution was obtained. For each combination, the mean values and variances were calculated. Figure 1, for example, illustrates the obtained results for F1 function optimization with the C_{11} combination.

IV. Results

To evaluate the SGA performance and to compare it with the results found in literature, the following values were used:

- population size: 50 individuals;
- number of generations: 80;
- number of runs: 100;
- chromosome or bit string length: 8 bits per solution;
- crossover mechanisms: single-point and uniform;
- crossover probability (p_c): 0.6;
- mutation probability (p_m): 0.001.

Considering the benchmarking function F1, after the entire tests, the obtained results to the combinations set are depicted in Figure 2. Table I summarizes the numerical values. The criteria adopted to measure the performance of best individuals were made by measuring the mean value

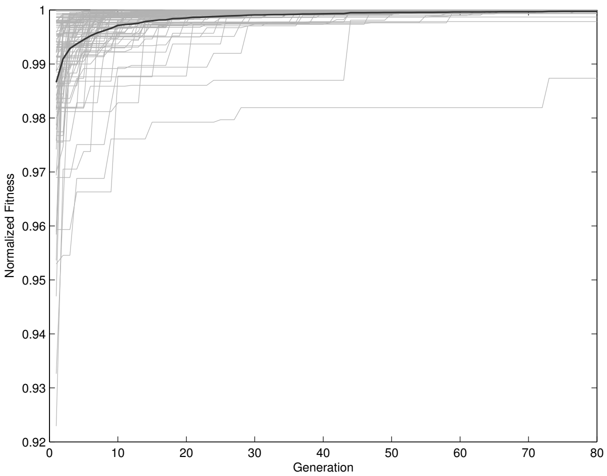


Figure 1. 100 Independent runs and the mean value to F1 function with combination C_{11}.

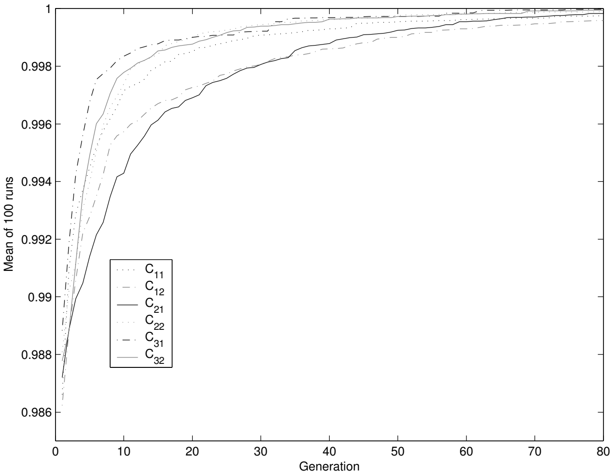


Figure 2. Comparative of the mean of the normalized fitness, to F1 function, in 100 runs.

Table I
MEAN VALUES AND VARIANCES TO F1 FUNCTION.

	10th generation		80th generation	
	μ	$\sigma^2 \times 10^{-5}$	μ	$\sigma^2 \times 10^{-5}$
C_{11}	0.9972	1.5339	0.9997	0.1644
C_{12}	0.9957	2.3725	0.9996	0.0454
C_{21}	0.9943	3.9221	0.9998	0.0167
C_{22}	0.9974	1.7816	0.9999	0.0249
C_{31}	0.9983	1.3449	1.0000	0.000119
C_{32}	0.9978	0.8595	0.9999	0.005012

and variance over the runs, becoming a criteria for numerical comparisons.

The values in Table I show excellent results to any combination for ongoing and stopped analysis. The low values of variance demonstrate that any run can be effective in the search of the optimal solution.

The obtained results to the combinations for functions F2-F5 that are depicted in Figures 3-6 and Tables II-V show the summarization of the results respectively by the measure of mean and variance, respectively.

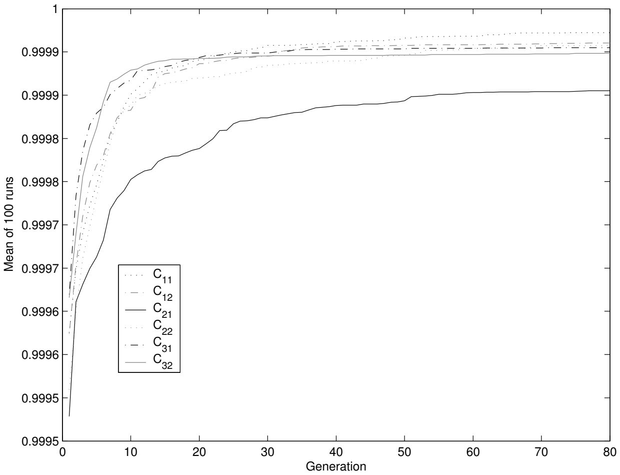


Figure 3. Comparative of the mean of the normalized fitness, to F2 function, in 100 runs.

Table II
MEAN VALUES AND VARIANCES TO F2 FUNCTION.

	10th generation		80th generation	
	μ	$\sigma^2 \times 10^{-8}$	μ	$\sigma^2 \times 10^{-8}$
C_{11}	0.9999	1.7021	1.0000	0.2462
C_{12}	0.9999	3.5903	1.0000	0.5344
C_{21}	0.9998	6.5557	0.9999	2.3654
C_{22}	0.9999	2.5488	1.0000	0.9630
C_{31}	0.9999	1.6755	1.0000	0.5718
C_{32}	0.9999	1.2254	0,9999	0.7725

Table III
MEAN VALUES AND VARIANCE TO F3 FUNCTION.

	10th generation		80th generation	
	μ	$\sigma^2 \times 10^{-4}$	μ	$\sigma^2 \times 10^{-4}$
C_{11}	0.9978	2.3391	1.0000	0.0000
C_{12}	0.9989	1.1815	1.0000	0.0000
C_{21}	0.9967	3.4728	1.0000	0.0000
C_{22}	1.0000	0.0000	1.0000	0.0000
C_{31}	1.0000	0.0000	1.0000	0.0000
C_{32}	1.0000	0.0000	1.0000	0.0000

In tables IV and V,the average decrease between rounds over the previous tables due to the nature of performance

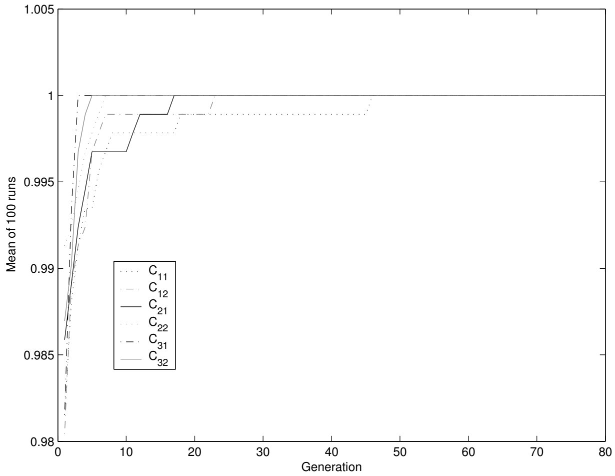


Figure 4. Comparative of the mean of the normalized fitness, to F3 function, in 100 runs.

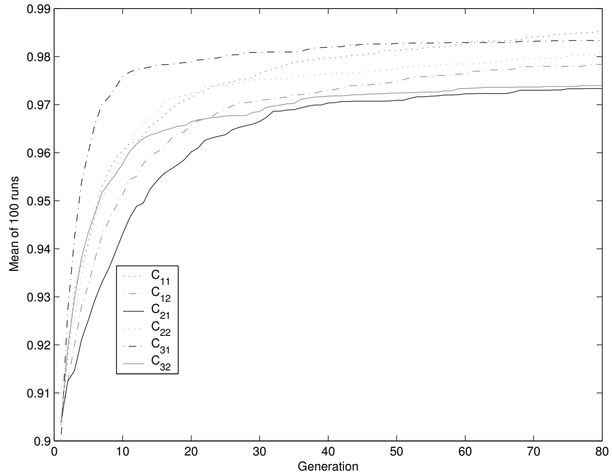


Figure 5. Comparative of the mean of the normalized fitness, to F4 function, in 100 runs.

Table IV
MEAN VALUES AND VARIANCES TO F4 FUNCTION.

	10th generation		80th generation	
	μ	$\sigma^2 \times 10^{-4}$	μ	$\sigma^2 \times 10^{-4}$
C_{11}	0.9604	7.1088	0.9853	2.2892
C_{12}	0.9512	9.1016	0.9784	2.9781
C_{21}	0.9431	13.123	0.9733	6.0680
C_{22}	0.9594	6.5950	0.9809	2.5381
C_{31}	0.9759	2.3428	0.9833	1.4531
C_{32}	0.9576	5.6891	0.9743	4.5210

among the results is verified. To elect a winning combination we decided to analyze the one variance with the lowest since the average of the same order of magnitude. For the two moments of evolution, the combination C_{31} is the one with the best performance among the results. The lower values obtained for the variance show that any round can be effective in finding the optimal solution. After analysis of these data, it appears that the solutions for all combinations of the set of operations can be considered as optimal solutions.

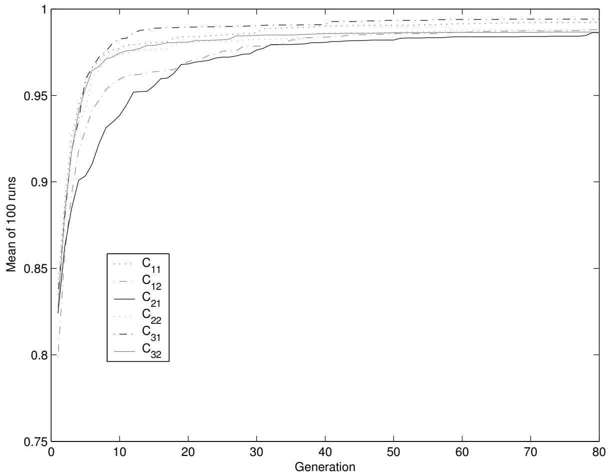


Figure 6. Comparative of the mean of the normalized fitness, to F5 function, in 100 runs.

Table V
MEAN VALUES AND VARIANCES TO F5 FUNCTION.

	10th generation		80th generation	
	μ	$\sigma^2 \times 10^{-3}$	μ	$\sigma^2 \times 10^{-4}$
C_{11}	0.9771	1.63	0.9923	0.8348
C_{12}	0.9593	5.14	0.9882	1.4573
C_{21}	0.9384	16.53	0.9863	1.6812
C_{22}	0.9736	2.87	0.9884	1.3815
C_{31}	0.9827	1,77	0.9941	0.5224
C_{32}	0.9745	1,13	0.9866	1.2796

After examining these results, the good results gotten in all the considered set's combinations are verified. To compose a generic sketch with all benchmarking functions and the entire combinations set, these results were combined. The Figures 7 and 8 show the similarity in performance to the test functions and the combinations.

These results show that the benchmarking functions F1, F2 and F3 impose the same behavior to all operators combinations and, consequently, reliable results are obtained. The functions F4 and F5, due to their nature, impose a different behavior to the operators combinations. In this context, the C_{31} combination preserves its good performance relatively to others combinations, and the differences in the values, verified according the variances values, are not significant.

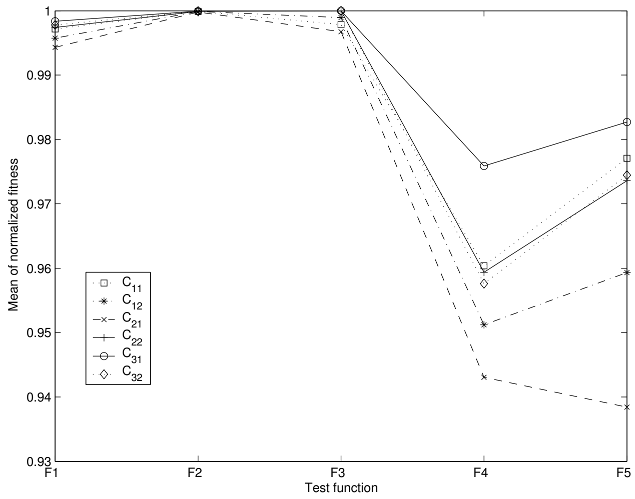


Figure 7. Performance analysis at 10th generation.

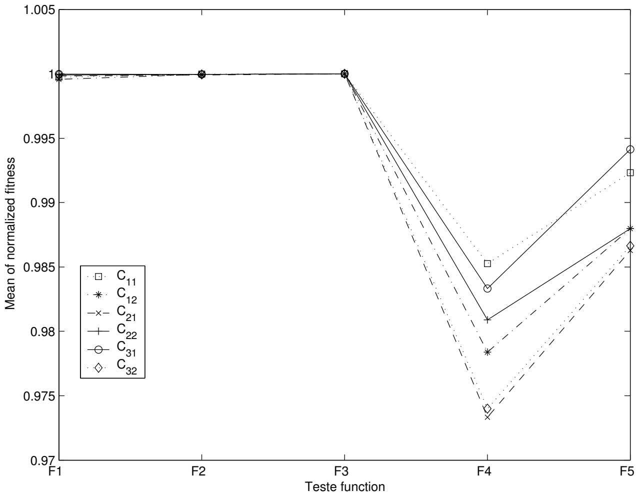


Figure 8. Performance analysis at 80th generation.

V. CONCLUSION

This paper has accomplished a performance analysis for the SGA approach - a simple genetic algorithm. This analysis, using a subset of benchmarking functions and doing combinations of operators, show the effectiveness of the SGA algorithm. In other words, given a search space, a convergence region is provided; local optima, which are widespread in some objective functions, are overcomed; and a useful set of feasible solutions is reached. These experiments show that the combination of stochastic tournament with single-point crossover is the combination that provides better results. The results described in this paper are significant because they show that the basic formulation of SGA is competitive in the different contexts found in the objective functions.

REFERENCES

[1] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learnin*. New York: Addison–Wesley, 1988.

[2] J.-Y. Xie, Y. Zhang, C.-X. Wang, and S. Jiang, "Genetic algorithm and adaptive genetic algorithm based on splitting operators," in *Jisuanji Gongcheng yu Yingyong (Computer Engineering and Applications)*, vol. 46, no. 33, 21 Sep. 2010, pp. 28–31.

[3] J. Digalakis and K. Margaritis, "An experimental study of benchmarking functions for evolutionary algorithms," *International Journal of Computer Mathemathics*, vol. 79, no. 4, pp. 403–416, April 2002.

[4] K. S. G. A. Jayalakshmi and R. Rajaram, "Performance analysis of a multi-phase genetic algorithm in function optimization," *The Institution of Engineers (India) Journal - CP*, vol. 85, pp. 62–67, November 2004.

[5] J. C. F. Pujol and R. Poli, "Optimization via parameter mapping with genetic programming," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. LNCS, X. Yao, E. Burke, J. A. Lozano, JimSmith, J. J. Merelo-Guervós, J. A.Bullinaria, J. Rowe, P. T. AtaKabán, and H.-P. Schwefel, Eds., vol. 3242. Birmingham, UK: Springer-Verlag, 18-22 Sep. 2004, pp. 382–390.

[6] M. Haseyama and H. Kitajima, "A filter-coefficient quantization method with genetic algorithm," pp. 399–402, 1999. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ISCAS.1999.778869, Last access date <retrieved: 10, 2011>

[7] J. Zhang, H. S. H. Chung, and W. L. Lo, "Pseudo-coevolutionary genetic algorithms for power electronic circuits optimization," *IEEE Trans. on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol. 36, no. 4, pp. 590–598, 2006.

[8] K. A. D. Jong, *Evolutionary Computation*. Cambridge, Massachusetts: MIT Press, 2006.

Rotation-oriented Collaborative Air Traffic Management

Udo Inden
Cologne University of Applied Sciences
Research Centre for Applications of Intelligent Systems (CAIS),
Cologne, Germany
e-mail: udo.inden@fh-koeln.de

Stephan Tieck
EADS Innovation Works
Research Team "The Future of Flying - Services, Maintenance & Logistics",
Hamburg, Germany
e-mail: stephan.tieck@eads.net

Claus-Peter Rückemann
Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster (WWU) / North-German Supercomputing Alliance (HLRN),
Münster, Hannover, Germany
e-mail: ruckema@uni-muenster.de

Abstract—Considering developments in aviation industry and ICT a framework of further steps of integration in Air-Traffic Management is drafted. Intelligent management systems and distributed parallel data processing is envisioned for including sequences of flights into collaborative ATM. This includes options provided by latest developments regarding the Internet of Things as well as of Things that Think, i.e. manifesting autonomous behavior in a complex operations environment. Advanced communication system components, verification methods, and mission critical communication networks are utilized to interlink distributed compute, storage, and High End Computing resources and create a fast, secure, and reliable system and operating environment.

Keywords-Air Traffic Management (ATM); aircraft rotation; complex operations system, intelligent organization; "Things that Think", operations flexibility; real-time management; criticality management; High End Computing; distributed resources; high performance communication networks.

I. INTRODUCTION

Aviation operations systems are of considerable complexity and facing an enormous growth of traffic. The SESAR program (Single European Sky Air Traffic Management Research [15]) prepares for a shift from central control towards a stepwise deployment of a self-organizing and intelligent managerial organization which will include the "intelligent aircraft". Future Air Traffic Management (ATM) employs intelligent functionality to track, maintain or if need be recover the plan of flights by hedging it against unplanned events respectively to coordinate its recovery.

Complex operations are marked by large numbers of unexpected, thus unplanned events and *we define intelligence as the ability of a system to targeted and timely response* [5]. But any response needs to re-allocate resources or to employ additional ones – i .e. it needs flexibility in terms of disposable buffers. If they are exhausted the system will be unable to capitalize on any intelligence.

Aviation systems are critical if delays of flights are about to propagate without chance to recover. Criticality [9] is a holistic property of a system in a point of time and likewise it needs a holistic and timely scope of management. For this, we suggest extending the flight-oriented scope of ATM by establishing an additional focus on aircraft rotations. *Rotations are sequences of flights of an aircraft in a period of time (e.g. a day)* and with this include the chance of managing interdependencies of flights. For this we suggest using the capabilities of future "intelligent aircrafts" [1] developed under the regime of the SESAR program. Accordingly we consider examining potential contributions by the upcoming internet of things (IOT) and of "things that think" (TtT) being equipped for some autonomous reasoning and collaborative decision making. [6] [7]

Pursuing this approach among others requires deeper research on the control parameters of aviation systems and on appropriate IT-support particularly with respect to potential contributions of intelligent things and distributed processing of data. Further systems are to be tested effectively integrating communication technologies and the world of things.

The paper provides an initial framework of intelligent real-time ATM applications based on the convergence of growing challenges, current improvement programs and novel developments of ICT. ATM programs and basics of aircraft rotations are explained. Sections four and fife sketch proposed architecture and scenarios of intelligent, rotation-oriented ATM and discuss benefits of managing system criticality. Section six structures related computing strategies. Finally conclusions and an outlook are presented.

II. ADVANCING AIR TRAFFIC MANAGEMENT

With ten-thousands of flights, starts, landings and related ground operations, with millions of passengers (and shipments) air traffic forms a massively distributed system of actors (pilots, air and ground controllers, ground and terminal service providers, customers, etc.), highly interdependent but each deciding with at least some autonomy and with only local, thus limited information.

At the same time raising urgencies of environmental concerns or the growth of travel by 8.8 % and of traffic by almost 5 % annually [2] systemically aggravate the mix of problems, while tight public budgets and political constraints to extend infrastructures and small profit margins in the industry limit options to respond.

This system inevitably emerges floors of interference, interacting with external events (weather, security …) and preparing the ground for butterfly-effects. E.g., one ownerless suitcase can block a terminal and delay many flights. Along operational relationships then quickly service failures propagate across continental and intercontinental networks.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

Thus maintaining planned services or – more general – safe and efficient operations requires material and (close to) real-time management capabilities. An obvious strategy is to advance traffic management for a more efficient use of resources: With the (political) objectives to accommodate a threefold of current traffic, to improve safety by a factor of 10, to reduce environmental impacts by 10 % and to cut ATM costs by 50 % large and coordinated joined public-private undertakings have been started in all major aviation areas, e.g., SESAR (likely to go into its deployment phase by 2014) in Europe and NextGen ATM in the US [2][3].

Future systems include GPS-based control of 4-D flight trajectories, system-wide information management (consistent undelayed data sharing, improved proceedings and algorithms) or a higher degree of automation of control and of procedures to stabilize or recover flight plans. As a major advance aircrafts will get more choice in choosing routes rather than being limited to air-streets. With further improvements and supported e.g., by advanced Airborne Collision Avoidance Systems (ACAS) spatial separations of aircrafts will be agreed by peer-to-peer principles: Therefore the "intelligent aircraft will be a critical element in 21st century ATM." [3].

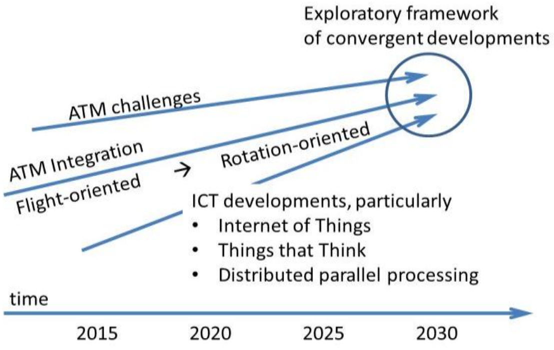


Fig. 1 Converging Developments

These programs have horizons of implementation of 10 to 15 years. In this time and after we assume three developments to converge (Fig. 1): Challenges to ATM will continue to increase, ICT will provide new options to answer and the current generation of ATM innovations will be implemented. With these developments reasons, technology and organizational concepts to integrate a wider scope of ATM converge. And with distributed parallel processing also issues of Grid and High-End Computing are touched. In the following we try an initial framework of these applications scenarios.

III. Management of Aircraft Rotations

Rotations (figure 2) cover deeper interdependencies between flights which cannot be managed on the level of individual ones. The intrinsic complexity of aviation systems – materializing in the form of failure propagation across networks – emerges on the level of rotations which also triggers the complexity of individual flight operations. This shall be explained on the example of scheduled airline services.

Rotations are planned in answer to the demand for transportation between origins and destinations in terms of its volume and distribution in time (daytimes and frequencies of service within a period), to connecting flights (e.g., in hub-and spoke networks), to distances (flight-time) or to availability of slots at airports as well as to the load-factors of aircrafts (utilization of a given fleet of aircrafts). Rotations include a number of legs (flights). I.e., problems which have occurred in the first leg may affect subsequent ones. In case of transfer connections the problem may also propagate to rotations of aircrafts operating connected flights. And with aircrafts also crews move in networks, air-craft maintenance is planned or many inventories of equipment distribute.

Operations footprints – in terms of direct (variable) costs, resource and infrastructure utilization (fixed costs), environmental efficiency (emissions, consumption of water) depend on the efficiency of rotations. For example maintenance footprints are to be managed on that level. While rotations are efficient if all flights are efficient it is not true that maintaining the efficiency of individual flights automatically maintains the efficiency of their interdependencies and inefficiencies are likely to accumulate high expenses.

In competition airlines need to manage conflicts between aircraft utilization asking for short(er) ground times (aircrafts only make money when they fly) and service quality by trend affected by such measures. If propagation of service failures turn into dissatisfaction of customers an airline with e.g., 100 aircrafts, each in average rotating in a network with six legs per day, has to calculate whether and how 5 minutes more ground-time for each of the 600 legs = 3000 minutes = 50 hours (equal to the average employment of 3 aircrafts) of idle time is paid by the avoided annoyance of passengers.

Legacies accumulating in operations systems are the second major driver of operations complexity. And in the multi-national, very political, hierarchically organized and for good reasons also very risk-avers world of aviation change takes time and time produces "renovation holdups". Tidying up such holdups is the core of lean-management programs. Since current management principles have been settled shortly after World War II. SESAR or NextGen are of that type. In front of this background a paradigm shift is inevitable (and a challenging change program).

Lean-management programs can provide a respite. But stress will return and the "granularity of object and time" [4] will increase: An ownerless suitcase may block a terminal, a late push-back, a defect stair, literally any disorganized resource may ruin rotations. And under stress it makes a difference whether resources are planned offline and "the next free one" is ordered to service or whether they are continuously tracked and planned online to events. I.e., there is reason considering the integration of the layer of flight rotation into real-time service maintenance. This layer will interact but not interfere with ATM concerned with the flights.

A threefold of current traffic will be reached in 15–20 years, soon compared to the time it needs to realize new airports in Europe. Thus any large airport is under *continuous* physical as well as organizational re-construction. In the words of a senior manager of a large European one: "We are evolutionary driven." Thus if flights and not rotations are the organizing principle of control achievements of SESAR or NextGen are easier to be consumed by growth or by competition (for example cost cutting, service / quality increase).

IV. On the Architecture of an Intelligent ATM

The capability of timely adapting activity to unexpected change (intelligence) emerges from the capability of acting units (agents) in a distributed system to freely associate or re-associate in a context which establishes interdependence for a certain time. A rotation is an example of such a context. I.e. objectives are achieved by autonomous re-allocation of resources or – if solution space is exhausted – to relinquish minor objectives for maintaining superior ones.

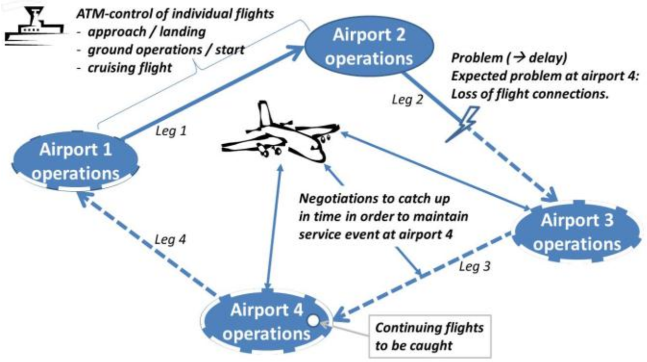


Fig. 2 High-level Model of Rotation-oriented ATM

Ideally, all agents are satisfied with their individual plans and plans of all agents are non-contractively coordinated. Then an unplanned event may cause, that at least one agent has become unable to achieve its objectives and now will try to improve again by re-negotiating its contracts with other agents. I.e., dissatisfaction propagates along services asked from respectively provided to other agents until a new satisfying solution is achieved.

This depiction equals design principles of a multiagent system emerging intelligence from relation-based interactions by associating to each other accordingly to the fit of properties (e.g., need, capability, objectives) as described in ontologies (structured domain knowledge). Market-based coordination is a promising approach. [5] However we do not suggest developing another mirror-MAS for simulating or managing a real operations system.

A. Collaboration scenario

Rather we consider including all relevant stakeholders, whether persons, objects or systems concerned and taking a relevant role in the current operations context by inviting them to a "Service Maintenance Conference" (SMC) taking place in an appropriate network environment (a private, proprietary cloud). It compares to an on-demand conference call which is organized accordingly to MAS principles. Figure 2 gives a basic idea who or "what" may participate:

- The aircraft takes the chair of the conference, possibly supported by the airline flight-center,
- Flight operations:
  - Other aircrafts in the operations vicinity if the current trajectories of the aircrafts are affected,
  - ATM authorities supervising or directly controlling en-route and particularly near-airport ("terminal") flight operations,
- Airport ground operators at airports relevant in the context, particularly subsequent ones in the rotation:
  - Airport control centers.
  - Dispatch centers of ground services for aircrafts, in case also around passengers or cargo.
- Potentially further stakeholders of operations.

In a rough estimate about 100 – 200 instances may be included. Currently airport ground services are only coupled to ATM via flight plans and flight plan updates. In the model we propose to actively engage them in solution finding. The way this inclusion will be organized depends on the vision of the ICT in 10 to 20 years from now.

B. Scenarios and Trends to be Considered

SESAR architectures rely on SWIMs and consistent re-planning. In case of unexpected events stakeholders are responsible to take action accordingly to standard procedures, in future supported by systems developed by SESAR. Thus as a first step this new ICT is to be connected into a peer-to-peer network forming a second layer of ATM which interacts but not directly interferes with the first layer: flight management.

Yet there is another trend, marked by the visions of the Internet of Things [6] respectively of Things that Think [7] e.g., the next generation of aircrafts. The Car-2-Car Communication Consortium [8] is a further example aiming among others at avoiding accidents or the exchange of route information. At airports apron field vehicles (push-backs, tank- or de-icing trucks) will manage their activity. Dolleys (transport carts) will be RFID tagged, motorized boarding stairs with GPS and suitcases be equipped with tags remembering owners not to leave them behind.

In 2020+ not only aircrafts but most critical resources at airports will be able of some autonomy; almost any other will be at least connected. Directly or non-directly they will be able to participate in SMCs.

C. Concluding Scenario

This "internet of things that think" develops because ICT makes it possible and affordable with cheaper and better performing hard- and software. But the fundamental driver is different: Managing under conditions of high complexity and dynamics implies that more single objects become source or subject of events [4] and that reliable, correct and immediate information matters. In these environments centralized control fails and not at least "things" will obtain autonomy. They are equipped with sensors and with capabilities of reasoning or become users of the internet because this enables to capture and exploit first-hand information.

This is summarized in the sigh of a dispatcher "If I would know where it is!". The timely and correct answer makes the difference between a solution and a service breakdown.

Thus, the second scenario is that not just some hundred but thousands of "things that think" will participate in SMCs and because of the interdependence of rotations of aircrafts in several SMC in parallel. The result is a heterogeneous holonic network of ad-hoc SMCs and invitees include complex ATM systems, less complex dispatch systems for ground services, human agents like air controllers and service dispatchers and things: the aircraft and likely many others.

ISBN: 978-1-61208-161-8

V. Benefits: Management of System Criticality

In the scene drawn in Figure 2 interference occurs in the second leg, hours ahead of the event planned at airport 4. Since there is obviously plenty of time to organize response offline - what is the benefit of (almost) real-time re-scheduling in this case? Actually it is the "criticality" of the system, a major control parameter of managing complex systems.

A. Benefit of Real-Time Maintenance of Rotations

Effective response to unexpected events implies that (1) non-expected states of operations occur (and that respective information is valid), and that (2) the system is "intelligent", i.e., able of finding a solution which (3) can be physically implemented: There must be leeway to re-allocate resource. Therefore flexibility (buffers, slack, or redundancy) is the "raw material" of operations intelligence.

But flexibility is a volatile resource. In this moment it is available, in the next it is not: I.e., decisions made in the scene of Figure 2 are bets on the flexibility available 10 hours later. And finally flexibility may be "out of stock".

The benefit of real-time maintenance of aircraft rotations is hedging these bets over the time left – and with this the overall efficiency of the system! There is no steady state in aviation operations. There is constant change only. Even if all internal parameters are controlled (very unlikely) there are enough external ones. Given a threefold of current traffic thousands of maintenance conferences will run in parallel and the "flexibility status" of the system will fluctuate.

B. Criticality

Criticality is a decisive control parameter of managing a complex system. The concept has been coined in physics where it defines the scale-free point of a phase transition (e.g., from liquid to solid) or the transition from stability into instability of a pile of sand or of snow forming an avalanche. [9] In the meantime this concept has been adapted by many sciences, among others in economics, in history science [10] or in business [11].

As a result from experiences and case studies, in the context of the new concept we can define:

- criticality as phase of transition amid capability and incapability to act due to exhaustion of flexibility.
- Intelligent real-time service maintenance (organized in SMCs) as a tool to actively manage criticality.
- the role of the aircraft as a 'supervisor' of criticality management with respect to rotations' efficiency.

The parameter of criticality focuses intelligence on the system-wide management of the most critical resource: flexibility (for an example see [12]). It is to be expected that this holistic approach combined with aircrafts which actively manage their rotations and related interdependencies will increase the economic efficiency of the overall system.

C. Focus of further Research

Resources of aviation systems are massively distributed in terms of functionality, space, organization or time – as flexibility is: Successfully responding to unplanned event may require coordinated, timely action of resources providing different functionality at different places, controlled by different organizations. In accordance with the market-based multi-agent approach explained above managing criticality equals managing liquidity (the volume of circulating money) by a central bank. This induces questions like

- How can flexibility be measured and criticality be estimated?
- Do operations' processes and performance (costs, quality, resilience …) exhibit "typical" patterns?
- Can patterns of causal behavior be exploited respectively how will non-causal patterns be treated?
- How does the distribution of flexibility (e.g., buffers) affect measuring and managerial options?
- How can SMCs be efficiently organized and technologically facilitated?

Answers to these questions will have impact to a theory of augmented intelligent organizations and the theory of volatile resources and both research as well as implementation will have to rely on technological advancements.

VI. Computing Strategies

In the past, ATM has not yet been apprehended as a domain of High End and High Performance Computing. High Performance Computing can be essentially defined to make use of the current high end resources available for a specific purpose. Making use of these resources can help to solve the performance barriers for practical ATM implementations in a next stage of development.

There are many practical problems with ATM systems that require immense computing power, for example, solutions that have to consider a large number of mesh points or locations need a huge number of operations to be calculated. On the other hand, one single processing step can otherwise afford a huge amount of memory. Complexity and state of basic high end technological development so far refrained from considering HPC technologies and HPC resources for implementing the components needed. With the last year's improvements in understanding these complex systems, integrated High End Computing has got into the focus of development. From the past studies we understand:

- how to create a SOA concept for rotation management, compatible to the SESAR SOA ATM-model.
- how to create collaboration models for system and architectures management and operation.

The main research topics resulting include:

- Distribution, job allocation to distributed resources.
- Capacity constraints versus size and granulation of problem.
- Robustness of algorithms and overall systems.
- Security of information and computing.
- Management and operating of HEC and HPC networks and resources.

The research program being defined by these issues includes the main sections (a) collaboration, management, and operating issues, (b) trust in information and computing, and (c) robustness and criticality.

When an unforeseeable change within existing planning occurs, the triggering of schedule modifications events from airplanes will be a suitable means for improving capability to respond. For optimizing the processes and sequences it is necessary to develop a performance based approach using HEC and High Performance Computing (HPC) strategies.

A. Collaboration, Management, and Operating

The resulting conceptual work used is based on the experiences and case studies done within collaboration projects over the last years. Based on the collaboration framework operation and management can consider multidisciplinary collaboration and legal aspects and integrate Service Oriented Architectures (SOA) and Resources Oriented Architectures (ROA) as with the GEXI framework studies [13, 14]. With the common heterogeneous structure necessary to build networked systems naturally strengths, facilities, and capabilities of disciplines, services, and resources provider groups differ. Collaboration aspects are the basic requirement for efficient and reliable systems engineering and maintenance, especially with complex multidisciplinary distributed systems and algorithms.

Two general computing paradigms and derivative combinations are available for organizing and particularly for coordinating across SMCs envisioned: ground-based computing and mobile airplane-based computing. In both cases compute requests have to be scheduled. In the case of ground-based solutions, requests and data will have to be sent to a ground based computing infrastructure. In the mobile airplane-based case a request has to be scheduled in order to get up-to-date information from the ground-computing and do pre-calculation on-board. Both architectures are clearly defined by capacity computing requirements.

B. Trust in Information and Computing

The implementation for a mission critical logistics computing chain has to rely on fast broadband networks and a secure network infrastructure which first of all needs to be interoperable with standards defined by SESAR or NextGen. Information exchange can be handled by means of verification [14]. The implementation considers signatures from a Certification Authority (CA) and checksums. The communication network used for air-ground communication will preferably be based on a dedicated network, highly protected, among other things against intrusion and Denial of Service (DoS) occurrences.

C. Robustness and Criticality

In order to work out in-time compute tasks, communication and computation have to be completed within less than about five minutes wall clock time. Certified information transfer is the base for secure any reliable information system usage and computing. In any case with mission critical implementations of distributed computing and mobile components a fallback solution is essential, based on data replication and emergency procedures. As non deterministic aspects will reduce the robustness of systems, the problem size is reduced to problem cells with defined conditions, like resources consumption and wall clock times, so that safe fallback states will be available for ongoing operation. This will facilitate the application of control procedures and intelligent automation of required operational tasks.

D. Concepts and Requirements

Various strategies and technologies can be used to make practical use of integration with HEC resources. Current base for providing computing power are:

- High End Computing (High Performance Computing, Supercomputing).
- Distributed and Services Computing (Cloud Computing, Grid Computing, Distributed Computing, MultiCore and ManyCore technologies).

The requirements to exploit high end resources and mobile highly performant resources leads to interlink intelligent systems with High End Computing and Distributed and Services Computing resources, mostly for capacity computing purposes. Integration of information and computing systems is commonly implemented using framework interfaces. Therefore a modularization of interactive and batch access to resources is mandatory. The solution is based on flexibility with parallelization: Loosely and massive parallel computing can be achieved using dynamic event triggering and on the other hand MPI and OpenMP implementations.

Regarding data and information exchange there is a strong need for dedicated high end networks. As with the environment middlewares and modular facilities, like accounting and communication services are needed for practical operation. Essential system components have to be build on common standards. Network, system, and data security is most important for mission critical systems.

VII. Conclusions and Future Work

With the results presented in this paper we have shown that the new paradigm with rotation-oriented Air Traffic Management for extended collaborative ATM can be a solution for future economic management. High end computing resources, communication networks, and High Performance Computing architectures are used to deliver the compute power needed. The concepts on criticality do support the need for mission critical systems. Operating complex dynamical system architectures, computing and information system resources can be handled with flexible collaboration frameworks in order to achieve efficient complex systems integrating information system technology and the world of “things that think” as well. This will allow a flexible and economic change and risk management and sustainable operation concepts. Advanced communication system components, verification methods, and mission critical communication networks are utilized to interlink distributed compute, storage, and High End Computing resources and create a fast, secure, and reliable operating system environment.

The goal for the near future is to make use of HEC resources like cloud computing and future High Performance computing systems. High end resources can be integrated with information, communication, and logistics systems by creating appropriate interfaces and services. The systems supported will gain access to distributed computing and storage power not available locally under any economic aspects otherwise. The demands for High End Infrastructure as a Service and provisioning of services will lead to a definable level of reliability and quality (QoS, QoD, QoE).

For the next generation of large complex intelligent systems we need fully integrated network and component management solutions. Following the technology improvements of services implementation, the mid-term focus is the integration of "High End Computing as a Service". In the future, developments and concepts will focus on bringing this concept into life with industry scale systems.

ACKNOWLEDGEMENTS

We are grateful to all partners in past and current programs for fruitful discussions and collaboration in projects and project developments contributing to concepts presented in this paper. Particularly we want to name Airbus R&T Management, Hamburg, DE; KölnBonn-Airport, Cologne, DE; Modular Galley Systems GmbH, Elchingen, DE; Vienna University of Economics and Business, Institute for Transportation and Logistics Management, Vienna, AU; FGM-AMOR, Graz, AU; Controlling Chaos Technologies, Hannover, DE; Knowledge Genesis, Samara / Swisttal, RU/DE; Institute for Control of Complex Systems / Russian Academy of Sciences, Samara, RU; Manchester Business School, Centre for Service Research, Manchester, UK; Rzevski Solutions, London, UK; Intelligent Communications and Broadband Networks Laboratory (ICB-Net) Athens, GR; Westfälische Wilhelms-Universität Münster, Zentrum für Informationsverarbeitung, Münster, DE; European Legal Informatics Study Programme, Partner University: Hannover, DE; North-German Supercomputing Alliance (HLRN), We also owe our thanks to the German Federal Ministry for Economics and Technology as well as to the German Aerospace Center / Project Agency for financial and managerial support of related research projects.

REFERENCES

[1] C. Scherer, "Executive VP Strategy & Future Programmes", Airbus, quoted in D. Thisdell, "*Aircraft must be smart swimmers to make next-gen ATM work*", Flight International 14/01/11 News Article.

[2] P. Hotham, "SESAR & NextGen Working together for Aviation Interoperability", Presentation to the Royal Aeronautical Society Annual Conference, London, April 14th 2011; original source: ICAO, International Civil Aviation Organization, 2011, URL: http://www.sesarju.eu/news-press/news/sesar-ju-royal-aeronautical-society-annual-conference-815 [accessed: 2011-05-10].

[3] booz & co., "Final Report, Adequate and Innovative Funding Mechanisms for the Preparation and Transition to the Deployment Phase of the SESAR Programme", Sept. 2010, URL: http://ec.europa.eu/transport/air/sesar/doc/2-2010_09_28_funding_and_financing_of_sesar.pdf [accessed: 2011-05-20].

[4] M. Rauer, S. Karadgi, D. Metz, and W. Schäfer, "Real-Time Enterprise – Schnelles Handeln für produzierende Unternehmen", in *Wirtschaftsinformatik, Mai 2010*, download / purchase link URL: http://www.wirtschaftsinformatik.de/index.php;do=show/site=wi/sid=c51138b3f49d6d3de7ae7bf711bde1ac/alloc=12/id=2730 [accessed: 2011-05-20].

[5] G. Rzevski, "A practical Methodology for Managing Complexity". Emergence: Complexity & Organization - An International Transdisciplinary Journal of Complex Social Systems. Vol. 13, Nos. 1-2, 2011, ISSN 1521-3250, pp. 38-56.

[6] "Radio Frequency Identification and the Internet of Things", European Commission, Information Society, URL: http://ec.europa.eu/information_society/policy/rfid/index_en.htm [accessed: 2011-05-20].

[7] Things That Think, "TTT Vision Statement", URL: http://ttt.media.mit.edu/vision/vision.html [accessed: 2011-05-14].

[8] Car to Car Communication Consortium, URL: http://www.car-to-car.org/ [accessed: 2011-05-14].

[9] K. Christensen and N. Moloney, "Complexity and Criticality", Imperial College Press Advanced Physics Tests – Vol 1, 2005, ISBN: 978-1-86094-504-5.

[10] R. Cliver, "Tremors in the Web of Trade: Complexity, Connectivity and Criticality in the Mid-Eighth Century Eurasian World", The Middle Ground Journal, Number 2, Spring 2011.

[11] "Complexity-Based Crisis-Anticipation for Corporations, Investors and Policy-makers", Ontonix, 2008, URL: http://www.ontonix.com [accessed: 2011-05-20].

[12] U. Inden, R. Franken, "Final Report: CESSAR – Configuration and Evaluation of Service Systems with RFID", Cologne University of Applied Sciences, (Sub-Research project 2011-5-7 in collaboration with Airbus, EADS. Funding: German Federal Ministry of Economics and Industry, Program Management: PT DLR Cologne, Project Number: MT06005), (in publication).

[13] Rückemann, C.-P., "Integrating Future High End Computing and Information Systems Using a Collaboration Framework Respecting Implementation, Legal Issues, and Security", in *International Journal on Advances in Security*, 3(3&4):91-103,2010. ISSN:1942-2636, URL: http://www.iariajournals.org/security/sec_v3_n34_2010_paged.pdf [accessed: 2011-05-01].

[14] Rückemann, C.-P., "Envelope Interfaces for Geoscientific Processing with High Performance Computing and Information Systems", in Proceedings of the International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2011), DigitalWorld 2011, February 23-28, 2011, Gosier, Guadeloupe, France, pages 23-28. XPS, Xpert Publishing Solutions, 2011, ISBN: 978-1-61208-003-1, URL: http://www.thinkmind.org/download.php?articleid=geoprocessing_2011_2_10_30030 [accessed: 2011-05-20].

[15] http://www.eurocontrol.int/sesar/public/standard_page/overview.html

Warfare Simulation and Technology Forecasting in Support of Military Decision Making

Eric Malmi, Ville Pettersson, Sampo Syrjänen, Niina Nissinen,
Bernt Åkesson, Esa Lappi
Defence Forces Technical Research Centre
Electronics and Information Technology Division
P.O. Box 10, FI-11311 Riihimäki, Finland
e-mail: eric.malmi@gmail.com, ville.h.pettersson@gmail.com, shsyrjan@gmail.com, niina.nissinen@mil.fi,
bernt.akesson@mil.fi, esa.lappi@mil.fi

***Abstract*—We describe a methodology that uses warfare simulation, data farming and technology forecasting in support of military decision making. Our approach explores the vast space of parameters regarding unknown properties of future weapon systems and other uncertainties that affect the outcome of a battle. Characteristics of successful outcomes are identified, providing insights to such questions as what kind of investments should be made to meet future challenges.**

Keywords-warfare simulation; data farming; stochastic simulation; decision support system.

I. Introduction

Technology forecasting serves two purposes in the military context: planning investments of future weapon systems and anticipation of an adversary's future capabilities. In the former, different weapon systems that will be available in the future are modelled and their contribution to one's own performance is analyzed. This analysis supports the decision making regarding what kind of investments should be made to optimize our future performance. In the latter, the adversary's possible future weapon systems are modelled and analyzed. This gives suggestions about preventative actions for meeting the future challenges.

To simulate the course of a battle on operational level, autonomous simulation software is usually insufficient. Instead, one needs to employ wargaming in which tactical decisions are made by a human operator and weapon system effects are simulated by the software, so called man-in-the-loop simulation. In this paper we focus on one such software which enables wargaming: combat modelling tool Sandis [1], which is developed at the Finnish Defence Forces Technical Research Centre. Sandis is based on probability calculus and fault logic analysis and can be used for comparative scenario-based analysis from platoon to brigade level. In Sandis, the player deploys the troops on a map and gives them movement and firing commands. As output, Sandis gives probability distributions of the unit strengths, operation success probability as well as a killer-victim scoreboard and medical situation average values.

The outcome of a wargame is influenced not only by the tactical choices made during the battle but also by the various simulation parameters. There may be variation in the parameters due to uncertainties in future technology or uncertainty in the decisions the adversary makes. In addition, we may want to study the effect of different actions and conditions or test how, e.g., the choice of ammunition type affects the results. These variations create a vast space of parameter combinations that needs to be thoroughly studied. It is often, however, impossible to calculate or play all the combinations. The concept of data farming [2], [3] addresses this problem.

The use of warfare simulation for evaluation of future weapon systems has been discussed in [4], which also included an evaluation of the applicability of Finnish simulation tools to such problems. Data farming has previously been applied to the Sandis combat model in [5], [6].

The methodology presented here bears some similarity to the approach presented in [7]. The approach presented in that paper was intended to support the planning of military operations, and consisted of constructing and evaluating possible futures, however, without applying data farming. The approach utilized various forms of warfare simulation as part of the evaluation process.

In Section II, we discuss the wargaming procedure. After this, in Section III, we present how the outcomes of battle variations are commonly analyzed and what kind of visualizations are used to support the analysis. Finally, in Section IV, we discuss the pros and cons of the methodology we have presented and future work.

II. Wargaming Procedure

In our method the battle is simulated in three phases. The first phase is the initial data farming phase, which consists of automatically computing several possible initial states of the battle from the initial state of the scenario. The second phase is the selection of representative cases, in which a small number of representative cases – say three – is chosen from the automatically generated initial states. In the third

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

phase these representative cases are played manually with one or more human operators making the tactical decisions. If we wish to study long battles, these three phases are repeated. The data farming phase begins whenever we need to study the effect of some uncertain parameters. The process in whole is illustrated in Figure 1.

A. Data Farming Phase

The data farming phase consists of choosing the different parameters we wish to vary, e.g., the accuracy of some weapon system, and then automatically simulating their effect. The vast number of possible parameter combinations can be explored efficiently with so called data farming methods [2], [3].

Data farming is the process of running a simulation several times over a large parameter space, and analyzing the simulation results for statistical trends and outliers. It is often impossible to model or predict complex real world phenomena accurately due to several uncertainties. Data farming addresses this problem by not trying to come up with a single definitive answer or prediction, but instead computing the entire landscape of possibilities in hope of understanding and gaining insight on the phenomenon.

Important elements in the data farming process are design of experiments, high performance computing, and analysis of results. The experimental design step includes choosing the appropriate computer models and the key parameters we are interested in. In the high performance computing step we run the simulation over several possible parameter combinations in a high performance computing environment. The analysis of results is often done by using standard statistical methods and visualization on the simulation output.

The data farming framework is not restricted to a particular simulation tool, any simulation software or computational model can be used. Several standard methods have been developed to facilitate the data farming process, such as the latin hypercube sampling [8], [9]. Latin hypercube sampling is a method that assists the parameter space exploration in the experimental design step. Instead of running the simulations for all possible parameter combinations, which is often impossible, latin hypercube sampling chooses a small subset of the parameter combinations with the intention that the subset covers the parameter space well. This reduces the amount of computational resources required without compromising the quality of results too much.

B. Selection-of-Representatives Phase

The data farming phase produces a very large number of simulated variations that can be used as initial states for manual wargaming. Ideally we would like to play each variation manually, but this is often impossible since manual wargaming is time and labour consuming. In the selection phase only a small number of representative cases is chosen for manual wargaming. The number of representative cases should be sufficiently small considering the available resources, but still large enough to cover the important aspects in the simulation results.

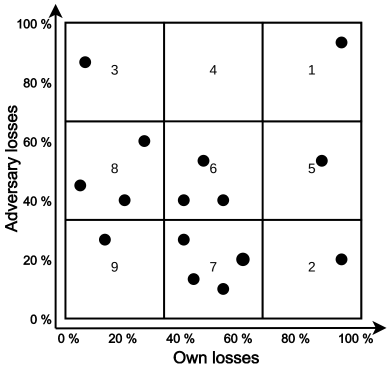


Figure 2. Illustrating the selection of representative cases by means of a scatter plot of adversary lossed versus own losses. Each dot corresponds to a simulation run with a certain combination of parameter values. The plane is divided into nine categories from which a few representative cases are chosen as initial states for the following wargaming phase.

Methods that are used to analyze and visualize data farming results can be useful in this phase. One way to choose the appropriate representative cases is to choose two interesting variables that describe the simulation results, and make a two-dimensional scatter plot of all the simulation results with these two variables as axes. In the military context the numbers of casualties on both sides are often the most interesting variables. The plotted variations can then be divided into nine categories as seen in Figure 2. Representative cases are chosen within those categories. One can eliminate uninteresting cases, such as the cases where both sides have been practically defeated.

As an example, consider Figure 2. Suppose that the x-axis represents own losses and the y-axis represents enemy losses. The cases have been divided into nine categories. Categories 1 to 5 do not need to be manually played, since in those cases at least one side is already defeated. Category 9 only contains one case, which is close to the cases in category 8. The representative cases can be chosen from categories 6, 7 and 8.

C. Gaming Phase

Once the initial parameters have been set, the gaming phase starts. In the gaming phase one or several scenarios are played out. The scenarios are derived from threat models and assumptions on how each side will use its forces and weapon systems [4]. Simulation tools are used to provide estimates of how each scenario will unfold. The scale of the combat analysis determines which simulation tools are suitable. One can distinguish between technical level, combat technical level, tactical and operational level analysis. For tactical and operational level analysis, man-in-the-loop simulations are often necessary. These simulation tools are based on tactical

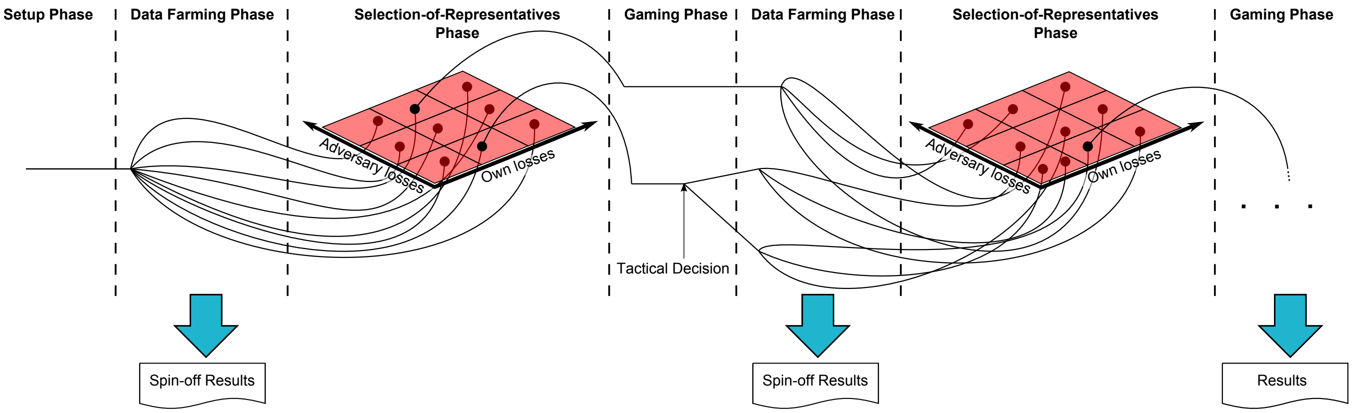


Figure 1. Illustration of the wargaming procedure. The process comprises three phases, which are repeated as necessary.

and technical models, and they calculate the effects and status of battlefield systems, whereas tactical decisions are done by a human operator. The operator decides, e.g., where to move the troops and what kind of firing commands the troops will follow. Once the operator has defined the tactical decisions, the simulation software calculates the outcome of the battle.

We have applied this methodology mainly with combat modelling tool Sandis [1]. A central componenent of the Sandis software is a map interface for wargaming. The units are deployed on the map and given movement and firing commands as input. The strengths of the units are probability distributions and combat losses are modelled as loss probabilities, calculated using a collection of weapon effect models. Additional calculation models include models for radio communication and medical evacuation. A feature of Sandis is that the calculation is based on Markov chains instead of Monte Carlo methods.

Sandis is designed for platoon to brigade level combat simulation. This provides a suitable scale for a wargame. However, we could use high-resolution simulation tools for analysing details and transfer the results of these analyses to the brigade-level wargame, bearing in mind that such a multi-level approach is very labour intensive [4]. Examples of sub-problems, which may require high-resolution simulation, include tank duels, in which individual tanks are simulated, and sensor system evaluation.

Making tactical decisions is the labour consuming part of the gaming phase. If we wish to continue with a new data farming phase, it is possible to automate the varying of the parameters and running of the simulation, once the baseline scenario has been played. Sandis also enables the operator to go back and forth in the timeline after the calculation has been finished. If the operator detects that an unrealistic tactical desicion has been made in the middle of the battle after some parameter change, the command can be refined and the simulation can be recalculated from that moment onwards. Certain parameter values may demand changes in the tactics of either side. The operator may also be forced to manually model the effect of some system or event. This may be the case when the simulation tool lacks the computational model of a particular system.

Furthermore, at some point in the scenario, it may be necessary to branch the scenario and investigate, e.g., two paths. Such a decision point may be whether a defensive position holds or is overrun, which leads to two different end states in the scenario. It is up to the operator to identify such critical points and divide the scenario into a number of major steps. Based on the plans for each side, such critical points could be identified beforehand, as discussed in [7].

In Sandis the manually played wargame scenarios are stored as XML files, as are units and their equipment and parameter data for weapons and equipment. For the data farming phase, the fields corresponding to the parameters to be varied are changed in accordance with the design of experiments, either by directly editing the files with scripts or by creating new versions of the files. The set of scenario files is executed in Sandis and the results from each simulation run are output to plain-text files, which are processed in the analysis step.

III. Analyzing the Results

The wargaming and data farming phases produce a large amount of simulation results that can be analyzed to gain insight on the research question at hand. Some simple analysis methods were already mentioned in the description of the wargaming procedure, but in the final analysis phase the results can be studied more thoroughly and from many angles based on chosen decision criteria. One can use, e.g., cost effectiveness analysis. In this section we describe some useful analysis methods.

As mentioned earlier, one common analysis method is making a two-dimensional scatter plot of all the computed variations. The dimensions of the plot can be any two interesting variables, e.g., casualties on both sides. The dots corresponding to favorable results, i.e., results where

ISBN: 978-1-61208-161-8

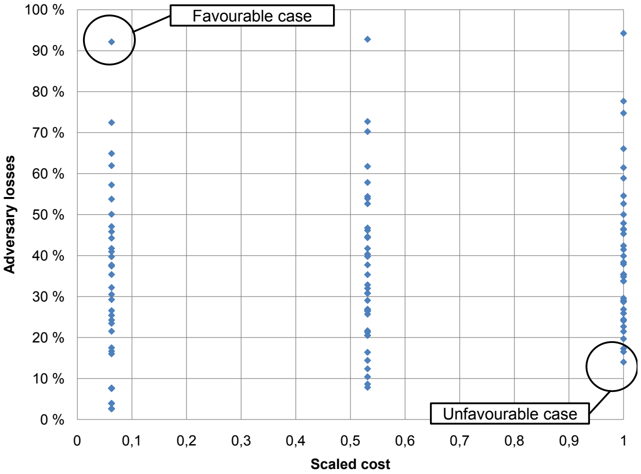


Figure 3. Scatter plot of adversary losses versus scaled own costs in a simulated scenario. Each dot corresponds to a simulation run with a certain combination of parameter values. In the lower right corner are cases where the adversary losses are small and one's own costs are high, i.e., the least cost effective cases. In the upper left corner, on the other hand, are cases where heavy losses are caused to the adversary for small costs.

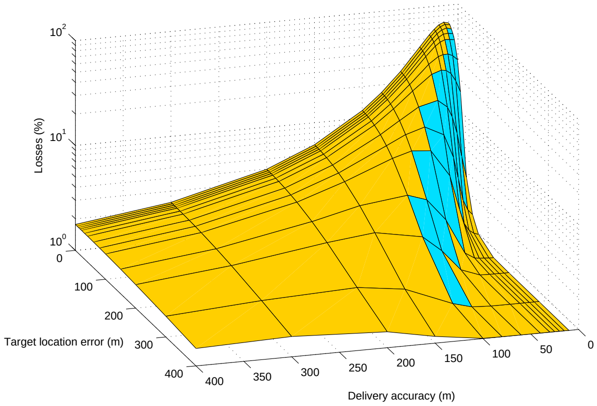


Figure 4. Example of visualization of data farming results. For a specific type of target element, i.e., an infantry soldier in hasty defence, the best indirect fire ammunition can be found. The delivery accuracy for the weapon is in this case expressed by the standard deviation in a circular bivariate normal distribution centered at the aimpoint. The target location error (TLE) is the difference between the true target location and the aimpoint. The colour of the surface indicates the most effective ammunition type for a given combination of delivery accuracy and TLE.

own losses are low and enemy losses are high, are studied in detail to gain insight on what made these cases good. Similarly, the results where own losses are high and enemy losses are low are studied to find out what should be avoided. Strategies and investment plans can then be improved to move towards the favorable results.

As an example of cost effectiveness analysis, consider Figure 3 that has been taken from real-world calculations. The x-axis shows the expected cost for an operation, and the y-axis shows enemy casualties. The values have been scaled to between 0 and 1. Each dot corresponds to one simulation run with some combination of parameters. Results from the simulations have been combined with cost information related to ammunition and equipment. Dots in the upper left corner correspond to favourable scenarios, where large enemy forces can be deterred in a cost effective manner. On the other hand, dots in the lower right corner correspond to unfavorable scenarios, where the enemy takes less losses despite the higher cost.

The parameter combinations and tactics that led to favorable outcomes can then be studied in detail. For example, one might find that some relatively cheap weapon system is effective when used properly while other more expensive systems do not perform well – a useful result when planning investments and strategy. Another possible result is to identify parameters, which have a strong influence on the outcome. The value of some weapon system parameter may have a large effect on the results, while the value of some other parameter may have a surprisingly small effect. These kinds of observations can be used to focus research efforts on improving the more important parameters of future weapon systems.

An example of how results from the data farming phase can be visualized is shown in Figure 4. From a data set of over 19 000 simulation results, generated through variation of several parameters, the most effective indirect fire ammunition against a particular type of target can be found with respect to two parameters: delivery accuracy and target location error. The delivery accuracy for the weapon describes the probability the ammunition hits a particular coordinate point and is expressed by the standard deviation in a circular bivariate normal distribution centered at the aimpoint. The target location error is defined as the difference between the true target location and the aimpoint. It is here treated as a systematic error, which is varied by moving the aimpoint.

It can be noted that the results from the data farming phase, besides providing a foundation for the wargaming, can be valuable as such and can be used to gain insight into the properties of weapons systems. Results like this are referred to as spin-off results in Figure 1.

IV. Conclusion and Future Work

A methodology combining warfare simulation, data farming and technology forecasting for military decision support has been presented. Using data farming the effect of system parameters on the outcome of the battle can be studied.

One downside of our methodology is that the wargaming phase is labour intensive, since it cannot be easily automated. This limits the number of different tactical alternatives that can be tested. The methodology is mainly intended for supporting peacetime aquisitions and planning when there is much time available. As noted in [4], evaluating different modes of operation for the adversary is crucial. Otherwise we might optimize the defence system for the wrong type of threat. Selection of scenarios for the analysis and selection of representative cases for a further gaming phase are therefore of importance.

We will continue to develop the methodology and incorporate additional features in order to handle a broader range of military problems. Although the wargaming process described in this paper utilizes only one simulation tool, the process can be extended to a multi-level multi-resolution simulation process, in which several simulation tools are used. After a specific detail has been analysed in a high-resolution simulation, the results can be transferred to a brigade level analysis tool, such as Sandis. This was discussed in [4]. Furthermore, when evaluating several branches of a scenario, it may be necessary, due to limited simulation resources, to play the most probable or most important branch thoroughly and the less important ones using a cruder simulation model [7]. Finally, although the domain studied here is land warfare, the methodology is applicable to all branches and the defence forces as a whole.

REFERENCES

[1] E. Lappi, "Sandis military operation analysis tool," in *2nd Nordic Military Analysis Symposium*, Stockholm, November 17–18 2008.

[2] G. E. Horne and T. E. Meyer, "Data farming: discovering surprise," in *Proceedings of the 2004 Winter Simulation Conference*, Washington, D.C., December 5–8 2004.

[3] G. E. Horne, "Summary of data farming," in *4th International Sandis Workshop*, ser. Defence Forces Technical Research Centre Publications, J. S. Hämäläinen, Ed. Riihimäki: Defence Forces Technical Research Centre, 2011, vol. 23.

[4] E. Lappi and B. Åkesson, "Combat simulation as a tool for evaluation of future weapon system and some risks in scenario based wargaming," in *Proceedings of the Conference Security in Futures – Security in Change*, ser. FFRC eBook, B. Auffermann and J. Kaskinen, Eds. Turku: Finland Futures Research Centre, University of Turku, 2011, vol. 5/2011.

[5] R. S. Bruun, J. S. Hämäläinen, E. I. Lappi, , and E. J. Lesnowicz Jr, "Data farming with SANDIS software applied to mortar vehicle support for convoys," *Scythe: Proceedings and Bulletin of the International Data Farming Community*, no. 9, 2011.

[6] J. Hämäläinen, P. Rindstål, R. Vuorsalo, R. Bruun, and J. Tiainen, "Using Sandis software combined with data farming to analyze evacuation of casualties from the battlefield," in *4th International Sandis Workshop*, ser. Defence Forces Technical Research Centre Publications, J. S. Hämäläinen, Ed. Riihimäki: Defence Forces Technical Research Centre, 2011, vol. 23.

[7] D. R. Pratt, R. W. Franceschini, R. B. Burch, and R. S. Alexander, "A multi threaded and multi resolution approach to simulated futures evaluation," in *Proceedings of the 2008 Winter Simulation Conference*, Miami, FL, December 7–10 2008.

[8] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, May 1979.

[9] T. M. Cioppa and T. W. Lucas, "Efficient nearly orthogonal and space-filling latin hypercubes," *Technometrics*, vol. 49, no. 1, pp. 45–55, February 2007.

A Flexible Suite of Software Tools for Medical Image Analysis

Alexander Nedzved
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
Nedzveda@tut.by

Valery Starovoitov
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
valerys@newman.bas-net.by

Abstract— **A new methodology of an intelligent software development for medical image analysis is proposed. The kernel of this software is a script interpreter that may be supported by an intellectual script generator. Commands of the interpreter are basic functions of image processing. The script generator forms new image processing scripts after analysis of image properties. This allows to change a processing sequence and the software interface. Software of such type may be self-transformed for different classes of medical images and various tasks in real time**

Keywords-medical image processing; intelligent software; script generator

I. INTRODUCTION

Computer engineering includes many different parts that influence to properties of software, for example, organization of functional support in the software affects the presentation of the user interface, and so on. Medical software has specifically requirements to software organizations. Such requirements depend of condition of solving tasks, knowledge of software user, and environment of user's work place. Therefore, the modern requirements to medical software is dynamic organization of functional, user interface and data managing for medical radiological methods, optical microscopy, endoscopy and ophthalmology, etc.

Any software may be divided into compiled programs and interpreters. Sometimes, software is represented as mixture of these variants. Such software for science has a compiled kernel that includes GUI (graphical user interface), basic functions and calculations, and different types of data representation. An interpreter usually is used as saved history of operations applied to a new image processing. In this case, the interpreter allows to create additional simple functions on base kernel possibilities but can not change the software [1]. We propose to use an interpreter as a kernel for our software. In this case the interpreter is used as a manager of action. It supports by performance of the functions, calculations, and GUI events. So, we can change the software design and organization by correction of interpreter scripts without a compilation stage. In other side, complex function and calculation are realized in external public compiled modules. This will keep the speed of calculation such as the compilation software. Today, similar software organization is used for web and game development. It is named an "open software architecture".

Our software is based on a mixture of a compiled library and an interpreter with many image processing functions. In result the software may be divided into two parts: for the professional software developers and the software designers or users. The interpreter has possibilities for including additional functions from a compiled dynamical library. It allows to change software properties and software applications without a compiling stage. On other side users can change the graphical interface for improvement comfort conditions for development and easy evolution of software. We apply such a technique for developing histology image analysis software [3].

Modern computer support and facilities in microscopy bring new perspectives in studying of cell structures. At the same time, the most commonly used method for a tissue analysis is still the well known morphological method, which allows to get reasonable biological conclusions after an image analysis. Group of morphological features, which are used for detection of similar types of cells and for organ and tissue fragment analysis, is noticeably extended. Usually, there is no any relation between different types of features. Therefore, types of histological tissue fragments are separated from each other by their morphological features. Systematization of histological objects is very important in order to provide a morphological analysis and oncologist diagnosis.

There are different approaches to segmentation of biomedical images. One of the most popular of them is based on mathematical morphology. Many morphology algorithms for cell segmentation have been proposed through the last years [3]. The initial image segmentation is determined by classifying the image local variation information obtained with dilation and erosion operations. A median filter may be used to smooth the segmented image. It removes small regions of misclassified pixels while avoiding significant changes to the cell profiles. The erosion operation is finally used to restore the cell areas. An edge-based segmentation may be divided into two independent stages: edge detection and edge linking. The obtained edges are used to determine the cell locations and contour model is further used to select the set of edges involved in the cell locations. Nedzved, et al. [4] have proposed an edge-based potential aimed at the elimination of local minima due to undesired edges. This approach integrates knowledge about features of the desired boundaries apart from gradient strength and eliminates local minima, which make the segmentation results less sensitive to initial contours. Color is an important feature in the histological image segmentation. There are several effective algorithms for automatic detection of cells and other histological objects [5]. However, these algorithms work under certain conditions to solve particular problems.

In addition, the new workplace is changing the software requirements. It is necessary to estimate functionality and compatibility of existed tools and data of software development to define a dynamical system of medical software developing. It has to be revised tasks, initial data, diagnostic features and characteristics.

The rest of the paper is organized as follows. In Section II we review properties of two different classes of medical images: histological and radiological ones. In Section III we describe a variant of software development which may be self-transformed after preliminary image analysis. In Section IV we describe testing of the presented software generation for various medical images.

II. FEATURES OF MEDICAL IMAGE ANALYSIS

In this section, basic features of image for determination of a way of processing are described.

A. Common processing sequence for image analysis of histological samples

Usually processing of histological images may be divided into several steps:

1) Input and image enhancement;
2) Segmentation;
3) Object detection (identification);
4) Measuring;
5) Analysis.

Every step consists of execution of a set of functions. Application of functions depends on properties or image estimations. It is possible to define such estimations in many cases, for example, for contrast, noise or blurring. We can construct table of image processing functions and image estimations.

For example, for histological images application of segmentation methods depends on many image conditions. Usually, an image is decomposed to separate regions to analyze the histological sample. Therefore, the segmentation process (i.e., extraction of homogeneous regions in image) is considered as a basic step for a formal scene description. It is necessary to define a correct set of features and feature characteristics for a suitable choice of segmentation methods.

Histological objects may be defined according to tasks of image analysis. Automated histological specimen analysis is based on topological features of images. It allows to define the whole procedure of study for object extraction. However, automatic analysis of histological specimen depends on the optical magnification of the image. In each magnification there is a certain group of topological features of tissue and its components. This fact has prompted to consider histological objects over magnification of histological specimens.

Fig. 1 presents a general scheme of hierarchical analysis of objects in histological images.

Different tissue fragments, which are composed of group of homogeneous cells and fibers, form an entire image of histological sample. Usually these fragments or objects are represented by a certain texture. Therefore, a region growing can be used for extraction.

From initial image conditions it is possible to define function for quality image processing and analysis (Fig. 2).

As a result, a table of connection function and image estimation are constructed.

In each step, functions are indicated by priorities. For example, for the image improving step the higher priority is defined for noise removal, next priority level contain a contrast enhancement and correction of the borders. Priorities determine the order of the functions and the need for re-analysis using neural network.

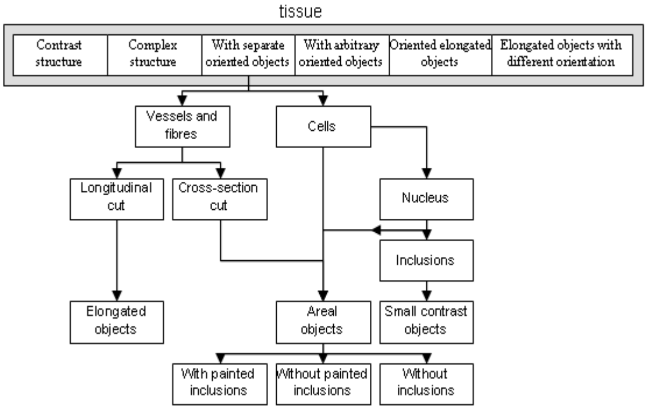


Figure 1. Hierarchical scheme of histological objects.

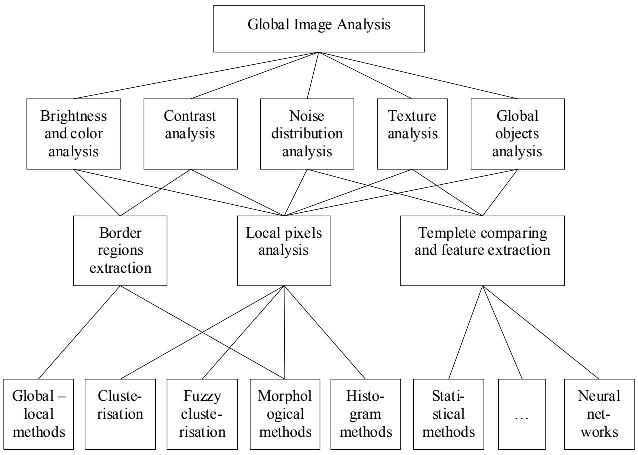


Figure 2. Scheme of segmentation methods definition from image conditions.

Image processing functions relate to basic computer vision topics. This is corresponding to its application for image changing. Every function changes properties of image and is applying for specific processing cases. Every function introduces in an interpreter table and can be supported by additional information.

B. Requirements for a radiological investigation software

Requirements for software are defined by a team of the basic users which will use achievements of this work. This team consists of physicians and medical workers of following specialists: the attending physician the oncologist, the physician of radiodiagnostics, workers of registry. They define the basic requirements to software They define the basic requirements of users to a complex.

Input, loading and presentation of images it should be carried out by means of following possibilities:

- Possibilities of operations with raster, vector formats (including DICOM);
- Possibilities of generation and presentation of synthesized 3D images on the basis of contours, which are prepared by processing 2D layers.
- Possibilities of generation of graphic reports documents.
- Presentation through system for visualization 2D and 3D images on the monitor screen, and also for reports.

The analysis and processing of medical images of tumors includes:

- Image improving;
- Interactive function for objects selection;
- Automated function for objects detection;
- Measurements and calculation characteristics from images.

The analysis and processing of angiography images and data are expansion of the previous requirements by specificity of objects - vessels and their network. In this case it is necessary to take into consideration morphological and textural features of vascular system.

The monitoring first of all require to the ease of general interface of a complex:

- Possibilities of synchronous work with different investigations of patients,
- The organization of storage of the information focused on many cases of patient,
- Possibility of preservation of data for the further statistical analysis.

The above described requirements of users form functional requirements. Functional requirements define functionality software. developers should construct it for users tasks. Functionality defines efficiency of working out. The efficiency increase is reached by:

1) The developing of modular system of interaction program modules with loading different functionally,

2) Using of ready software packages of the simple level of initial functions,

3) The software complex should include following modules:

- The global module of synchronisation including universal principles, structures and the data for providing interaction with other modules.
- Loadings of the digital information and management of processing and analysis of medical images technology;
- The automated allocation and the analysis of slice images (for example CT);
- Volume restoration of formation and definition of volume characteristics;
- Definitions of topological features of vessels for angiographic investigation;
- Statistical comparison of results of analysis of images for different time;
- Generation of graphic documents;
- Measuring and analytical functions.

In result, the software for the radiological methods of investigation should be accompanied by additional display capabilities, and presentation and image, not only preparing the general scheme of image analysis in histological methods.

III. DEVELOPMENT OF MEDICAL IMAGE ANALYSIS SOFTWARE

This section shows foundations of a flexible suite of software tools for medical image analysis that was developed. For elaboration of structural scheme of software basis interface an estimation of functionality and compatibility of existed software development tools were done. Tasks which may be solved with software to be developed, initial data, diagnostic features and characteristics have been observed. For software developing we use C-language from Microsoft Visual Studio.

Based on material posted by Guillaume Marceau [1], who in his study used parameters of 72 implementations of programming languages, and compared them to 19 special tests, prepared by the project "The Computer Language Benchmarks Game" as a kernel chosen interpreter LUA [6].

In the first case data are processed using temporary file, which allow to analyze records (images). In another case a transfer by calling a run-time library is performed (Fig. 3).

Our software is based on the interpreter of Lua language. It was elaborated as the main module based on an interaction of complex modules. It includes a graphical interface, global variables and image storage structure. Architecture of the graphical interface was carried out by linking the Highgui library [7] from the OpenCV package [7]. The image processing and function analysis are supported by OpenCV library but connection Lua with OpenCV are realized by Lua-binding interface for a connection function of OpenCV with Lua.

An image structure is determined by a module of graphical interface into OpenCV library which is responsible for visualization and representation of images. Headers of image structures are global variables-pointer of interpreter

Lua. They have special type - userdata. Userdata corresponds to a pointer in the computer address space. This module also includes image read/write functions, basic functions of image processing and interactive contouring. All interactive functions return values in the event block, which changes global variables of the interpreter. A simultaneous usage of several modules is required for tasks of monitoring space-occupying lesion. In this case an interaction has been performing by using global variables of the Lua interpreter and properties of the userdata type of Lua interpreter.

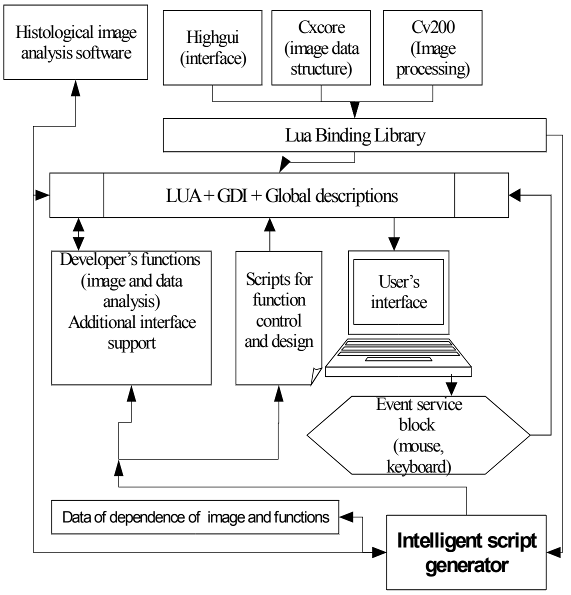

Histological image analysis software
Highgui (interface)
Cxcore (image data structure)
Cv200 (Image processing)
Lua Binding Library
LUA + GDI + Global descriptions
Developer's functions (image and data analysis) Additional interface support
Scripts for function control and design
User's interface
Event service block (mouse, keyboard)
Data of dependence of image and functions
Intelligent script generator

Figure 3. Scheme of flexible suite of software tools for medical image analysis.

Short sequences of functions are called in nodes for solving basic problems. During image processing a node of Lua can provide image exchange through Lua-outlets. First it sends images to the highest Lua-outlet identificator. After that the processing is done if the state of Lua-outlet was changed. The node changes the sequence of the processing functions and their parameters in accordance with his notifications. Loop notifications are sent eventually from a slow thread at the end of the scripts processing. The observer notification is required during the path through the each of the nodes. It is called the notify event-slot. This slot calls node's state method to get a dictionary with the node's attributes. Also it sends these attributes to the observers.

Communication between nodes in the same script is done through a direct function. This function is called by event-slots, either using a virtual method bang for the first inlet or a function pointer for other inlets. Images are processed by reference.

Adding new complex functions of processing and analysis is carrying out by group of developer's function. As an input new functions may get any global variable or text and numerical constants from the LUA interpreter.

All internal controlling of software is carrying out by text scripts of LUA, which are divided into two categories:

1. scripts for image sequences analysis;
2. scripts for operative functionality and setting up of software at workplace of medics.

All scripts are stored in a text form and easily accessible. However they are not for changing by user and may be edited for by developers only.

Scripts manage to software organization and create new additional function for image analysis and processing. Preprocessing analysis of images sets of image allow to define processing functions. Module of scripts generation defines such sets. It includes intelligent components for connections results of image functions processing and image characteristics. Of course such task can be solved only for particular task in our case for histology image analysis.

Every interpreter defines function through a specific table. We use it for definition connection images characteristics with a function in our software. It includes a set of feature vectors and variable of priority. The set of feature vectors defines the utility function. The variable of priority determines the position of function in the generated script. In result such software has intelligent self-programming possibilities.

In the analysis the first step is determining of global characteristics of images and defining the type of image. Single-channel grayscale image often correspond to the radiological methods of investigation in medicine. If the depth of the pixel brightness over the eight-bit image is defined as CT scans. The color image is composed of three or more channels. It is defined as histological. It remains uncertain class for eight-bit grayscale images, which can be classified as partially processed images of histology and radiological medicine. The difference halftone radiological images from histology most often lies in the way of formation of objects in images. Color histological image is formed on the basis of color in the preparation of amino acids. As a result, objects in an image composed of small specks, which form additional local boundaries. The boundaries correspond to local extremes of the brightness intensity. Therefore, a Sobel filter is performed to determine such characteristics (Fig. 4). The result is a gradient image. The distribution of brightness in this image is similar to a Gaussian distribution.

Based on the characteristics of the asymmetry and the eccentricity of histogram is determined type of images, that is belonging to the histological or radiological class.

The image type defines processing scripts and user interface. System generate separate user interface on base type definition for histological images (Fig. 5) and radiological images (Fig. 6).

A script generation module consists of two parts: image analysis and script construction. The fist part started from global image analysis that include histogram analysis and basic statistical analysis for pixels distributions, fractal and texture analysis. On the base such analysis as estimation of noise, blurring, image characteristics are calculated. From the interpreter function table image preprocessing script are generated for image improving by such estimation.

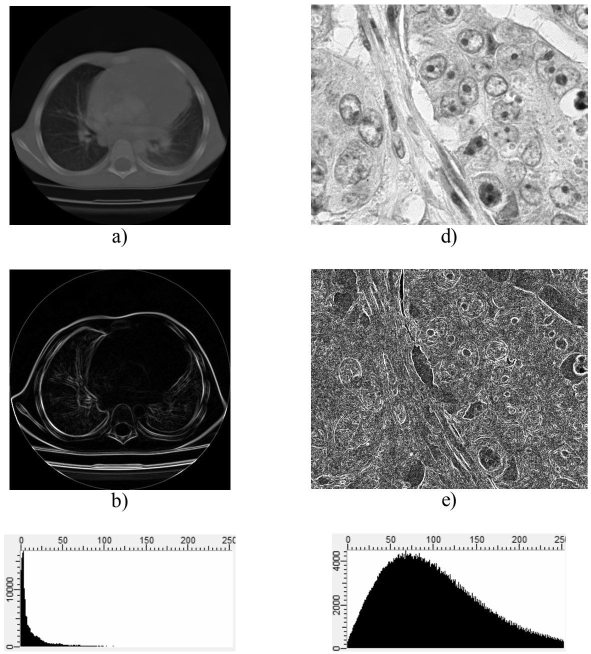
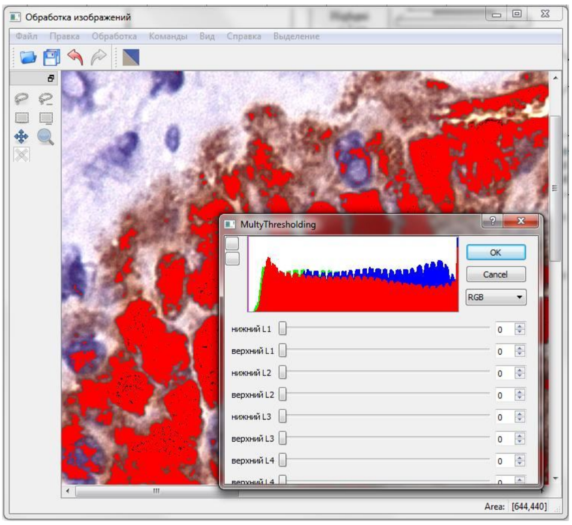
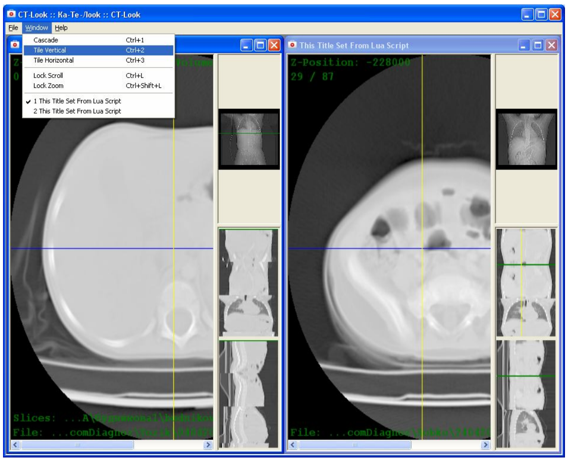


Figure 4. Examples of the results of image classification stages: a) radiological image, b) gradient of radiological image, c) histogram of radiological image gradient, d) histological image, b) gradient of histological image, c) histogram of histological image gradient.



нижний L1 0
верхний L1 0
нижний L2 0
верхний L2 0
нижний L3 0
верхний L3 0
верхний L4 0
Area: [644,440]

Figure 5. Screenshot of user interface generated for histological image analysis.


CT-Look :: Ка-Те-Лook :: CT-Look
File
Window
Help
Cascade Ctrl+1
Tile Vertical Ctrl+2
Tile Horizontal Ctrl+3
Lock Scroll Ctrl+L
Lock Zoom Ctrl+Shift+L
✓ 1 This Title Set From Lua Script
2 This Title Set From Lua Script
This Title Set From Lua Script

Z-Position: -228000
29 / 87

Figure 6. Screenshot of user interface generated for radiological image analysis.

Than local image analysis is going by convolution and statistical analysis of line-profile characteristics. This analysis allows to estimate characteristic of cells borders and contrast. Such estimation defines functions for image contrasting and border emphasis. After generation result image is tested for quality of processing. If the image quality is low stage of image analysis and function definition should be repeated. Such procedure generates image improving scripts that can be change by a user or developer. This script is only proposition and need to user control. The same mechanism of image analysis and function definition works for stages of segmentation and postprocessing.

As result our software constructs common script for every generation stages. Such script corresponds to function sequence for object extraction. It can be used for extraction of histological objects on the image. We use it for extraction of nuclei from histological images. Then characteristics of objects are calculated. It is necessary to detect type of objects that present at the image. We divide objects for five basic types: blobs, front, needles, dendrites and nets. Such procedure of object detection is spending by functions through script generation. Using global fractal, texture characteristics software detect geometrical type of objects and formed characteristics sets for object description.

For definition image processing function in scripts we try to use Kohonen neural network (self-organizing map, SOM) [8, 9]. It is a class of neural networks, the main element of which is a layer of Kohonen. The Kohonen layer consists of adaptive linear combiners. Estimations of global conditions on the image are used as weights in neural network. Adjusting of the input weights and vector signals quantization is closely related to a simple basic algorithm for cluster analysis (method of dynamic cores, K-means).

System is supported by script generator module. This module uses LUA-metatable of function for image processing and corresponding table with estimations of image. Thorough such table, definition of function sets going by Kohonen neural network. As a result, the module proposed scripts for image processing as text file. Users can spend analysis of this script and change some in it.

The software is supported by uses control and changing. There are a few version of user interface for managing of image processing and choosing of analysis type. Also the software has sets of interactive function for image processing. On the basis of these tests in the fifth section draws conclusions about the effectiveness of the proposed scheme, software analysis of medical images

IV. TESTING

Now, the software has stage of developing. But we spend a few tests for determination of basic possibilities.

This software was tested on different types of histological and radiological images. We are used three types of histological images and two type of radiological image (Tab. 1). Histological images were divided by cells density: high density - more than 70% image area for cells, middle density - between 40% and 70% image area for cells, low density - less than 40% image area for cells. Tested radiological images are divided into CT and MRI images. We spend tests for image improving stage. After testing was constructed table with probability values of success image processing by generated scripts..

A rate of success probability was defined by empirically way. The software was testes by 248 medical images.

TABLE I. TABLE OF SOFTWARE TESTING

Image type	**probability of type definition**	**probability values of success image processing**
Histological images with high density of cells	98%	83%
Histological images with middle density of cells	99%	90%
Histological images with low density of cells	87%	90%
CT images	100%	93%
MRI images	100%	95%

We take good marks for radiological images. For histological images marks are insufficient.

V. CONCLUSION

In this paper, we proposed a scheme of automatic generation of image processing function sets for analysis of histological tissue and described software for it. This software based on principles of open architecture and allows to change design and possibilities of it in real time on physician work place without compilation stage. In other side the speed of the program remains the same as in the compiled version. Developed software architecture simplifies the development of model programs for image analysis of histological images (Fig. 5) and radiological images (Fig. 6).

Marks that we take are unsatisfactory for histological images. It depends on a high complexity of images. We consider that it is necessary to change intelligent agent for script generator.

But in this paper we describe nice path for developing software for image processing. Basic possibilities of such software are possibilities of dynamical changing of interface and sequences of processing functions.

ACKNOWLEDGMENT

This work is supported by ISTC project #B-1682 and #B-1636.

REFERENCES

[1] G. Marceau "The speed, size and dependability of programming languages", Blog "Square root of x divided by zero", 2009 (http://blog.gmarceau.qc.ca/2009/05/speed-size-and-dependability-of.html, accessed 2011-06-06)

[2] G. Reitmayr and D. Schmalstieg. "An Open Software Architecture for Virtual Reality Interaction", proc of VRST'01, November 15-17, 2001, Banff, Alberta, Canada. (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.6256&rep=rep1&type=pdf, accessed 2011-06-06)

[3] T. Kanade, Z.Z. Yin, R. Bise, S. Huh, S. Eom, M.F. Sandbothe, and M. Chen "Cell image analysis: Algorithms, system and applications", Proceedings of WACV11, 2011, pp. 374-381

[4] A. Nedzved, A. Belotserkovsky, T.M. Lehmann and S. Ablameyko "Morphometrical Feature Extraction On Color Histological Images For Oncological Diagnostics", 5th International Conference on Biomedical Engineering, 14-16 February, 2007, Innsbruck: Proc. - P.379-384.

[5] L. He, L. R. Long, S. Antani, and G. Thoma, "Computer assisted diagnosis in histopathology", Z. Zhao ed., Sequence and Genome Analysis: Methods and Applications, ch. 11, iConcept Press, 2010. (http://www.iconceptpress.com/books/publicationContent.php?publication id=B00003, accessed 2011-05-28)

[6] R. Ierusalimschy "Programming in Lua", (second edition) Lua.org, ISBN 85-903798-2-5, p. 328, 2006

[7] G. Bradski and A. Kaehler "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly, p. 555, 2008

[8] S.. Kaski, "Data exploration using self-organizing maps", Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, Espoo, p. 57, 1997

[9] A Ultsch "U*-Matrix: a tool to visualize clusters in high dimentional data" University of Marburg, Department of Computer Science, Technical Report Nr. 36: 2003. pp.1-12

Digital identity-based multisignature scheme implementation

Francisco Javier Buenasmañanas Domíguez,
Ascensión Hernández Encinas, Araceli Queiruga Dios
Dept. Applied Mathematics
University of Salamanca
Salamanca, Spain
Email: {u61352, ascen, queirugadios}@usal.es

Luis Hernández Encinas
Dept. of Information Processing and Coding
Applied Physics Institute, CSIC
Madrid, Spain
Email: luis@iec.csic.es

***Abstract*—Digital signature, as an official signature, have many applications in information security, including authentication, data integrity, and non-repudiation. When a private or public document must be signed by a group of people, we call it multisignature scheme if all and every single member of the group signs the document.**

An identity-based digital multisignature is a multi signer digital signature so that the multiple private keys are generated by a trusted third part from signer's identities. In this paper, an efficient Java implementation to a recent identity-based multisignature scheme based on RSA is proposed.

***Keywords*-RSA; digital signature; multi-signature scheme; Java.**

I. Introduction

Adi Shamir [1] introduced a novel type of cryptographic scheme, based on the identity of the users, which enables any pair of users to communicate and sign documents securely. Moreover, it is possible to verify each other's signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party. The scheme assumes the existence of trusted Key Generator Center (KGC), with the role of giving each user a personalized smart card when he first joins the network. The card contains the secret key, and programs for message encryption/decryption and signature generation/verification. The user chooses any combination of name, social security number, e-mail or telephone number as his public key.

The scheme is adequate for closed groups of executives of a multinational company or the branches of a large bank or members of a sports club since the headquarters of the corporation can be the KGC that everyone trusts. This scheme can be the basis for a new type of personal identification card with which everyone can digitally sign checks, legal documents, and be electronically identify.

Moreover, a digital multisignature is a digital signature of a message or document generated by multiple signers with different private keys [2].

An identity-based digital multisignature, based on Shamir's identity-based scheme, is a digital signature of a message generated by multiple signers that obtains their private keys from a KGC, and the public keys from their own identities. Such practical and secure multisignature schemes were proposed to be applied for mobile communications [3].

In 2008, Harn and Ren [4] proposed an efficient identity-based RSA multisignature scheme and it seams to be secure against known attacks like forgerability under chosen-message attack, multi-signer collusion attack and adaptive chosen-ID attack. Authors propose the scheme with the most important multisignature properties: the length of the multisignature was fix and the verification time was also fix, regardless of the number of signers.

Some flaws on Harn-Ren identity-based RSA multisignature scheme were published a year later [5]. These drawbacks led to the proposal of a new system two years later. In fact, in [6] two security loopholes were discovered in Harn-Ren scheme and a new one was proposed. The resultant protocol was suitable for wireless communications because it is not only possessing security but also saving computation resources and communication bandwidth.

The implementation proposed in this paper supposes an efficient Java implementation of the improved identity-based multisignature scheme based on RSA suggested in [6]. This paper is organized as follows. Section II will detail the RSA cryptosystem and a short overview of existing identity-based multisignature schemes. Some attempts of using Matlab to implement these cryptosystems and the efficient Java implementation will be shown in Section III. Finally, conclusions and future works will be presented in Section IV.

II. Related works

In this section, we make a brief overview to RSA cryptosystem as well as Shamir's identity-based signature scheme. Harn-Ren's scheme as well as the improved multisignature scheme will be reviewed.

A. RSA cryptosystem

RSA cryptosystem [7] consists of three phases: key generation, encryption and decryption.

1) Key generation for a user U:

a) Select two large prime numbers p, q and computes $n = p \cdot q$ and $\phi(n) = (p-1)(q-1)$.

b) Select a positive integer e, $1 < e < \phi(n)$, such that $\gcd(e, \phi(n)) = 1$.

c) Compute the inverse of e in $Z^*_{\phi(n)}$, d, so that $e \cdot d \equiv 1(\text{mod } \phi(n))$.

The public key of U is the pair (n, e) and his private key is d. For security reasons, the values p, q and $\phi(n)$ must be kept secret.

2) Encryption process: If user B, wishes to cipher the message, M, and send it to user A, he carries out the following operations:

a) He obtains A's public key: (n_A, e_A).

b) He represents M as an integer in the range $[0, n_A - 1]$, even splitting m into smaller blocks if necessary.

c) The ciphered message is $c = M^{e_A}(\text{mod } n_A)$.

3) Decryption process: To decipher the cryptogram c and recover the original message, M, user A simply uses his private key d_A and computes $c^{d_A} \equiv M^{e_A d_A} \equiv M(\text{mod } n_A)$.

Asymmetric-key cryptosystems allow the sender to digitally sign a message, so that the receiver can check that the message is authentic and not modified.

Suppose that A, wishes to digitally sign a public document, M, and send it to B. The steps are the following.

a) The first step is to apply a *hash function* to the document, creating the document digest [8], $H(M) = m$, and encrypts it using his private key: $r \equiv m^{d_A}(\text{mod } n_A)$.

b) He ciphers the value r with B's public key to obtain the signature $s \equiv r^{e_B}(\text{mod } n_B)$.

Once the document and the signature are received by B from A, he can perform the verification phase as follows:

a) He computes $s^{d_B}(\text{mod } n_B) \equiv r^{e_B d_B}(\text{mod } n_B) \equiv r$.

b) He determines $r^{e_A}(\text{mod } n_A) \equiv m^{d_A e_A}(\text{mod } n_A) \equiv m$.

c) He deciphers c to obtain M, and checks whether the hash of M, $H(M)$, matches m. If it does, the signature is valid.

The security of the encryption and decryption processes, and the digital signature scheme based on RSA, depend on the difficulty of solving the factorization problem, which at present is considered computationally infeasible.

B. Shamir's identity-based signature scheme

First of all, each signer completes his registration with KGC, and KGC generates the signer's secret key using their own identities. On the other hand, the signature's public verification key is the signer's identification. This scheme reduces the costs of verifying the public key. The process is divided into the following phases:

1) KGC keys:

a) KGC picks two large primes p, and q, to compute n and $\phi(n)$.

b) Chooses a random public key e, satisfying §II-A conditions.

2) Signer secret key generation phase:

a) Signer j sends individual information and his identity i_j to KGC for registration.

b) After KGC accepts the user's identity, KGC uses his private key d to create a secret key $d_j \equiv i_j^d(\text{mod } n)$ from signer's identity i_j. Subsequently, d_j will be sent back to the signer as his secret key.

3) Signing phase: In the process of signing a document or message digest m, the signer uses his secret key d_j and the public key e of the KGC to produce the signature $\sigma = (t, s)$. The signing process is as follows:

Signers choose a random number r to compute $t = r^e \text{mod } n$. The secret key d is used to compute $s = d \cdot r^{H(t,m)}(\text{mod } n)$. Then, (t, s, m) is transmitted to the receiver, and $\sigma = (t, s)$ is the signature of the message.

4) Verification phase: When the receiver receives the signature $\sigma = (t, s)$ for the message m from signer i_j, the public key e of the KGC and signer's identity i_j can be used to verify the validity of the signature:

$$s^e \equiv i_j \cdot t^{H(t,M)}(\text{mod } n).$$

C. Harn-Ren efficient identity-based RSA multisignature

1) Private key generation phase: In this algorithm, every signer obtains his private key from the KGC:

a) Every signer sends their individual information to the KGC for registration.

b) KGC, with his private key d and the message digest of identity i_j, generates the private key $d_j \equiv i_j^d(\text{mod } n)$ of i_j signer.

2) Signing phase: To generate an identity-based digital multisignature every i_j signer from a group of signers, $i_1, i_2, \ldots, i_l$, follows these steps:

a) Chooses a random integer r_j, and with his public key, e, computes

$$t_j \equiv {r_j}^e(\text{mod } n).$$

b) Broadcasts r_j to all signers.

c) After receiving r_j, $j = 1, 2, \ldots, l$, each signer computes

$$t \equiv \prod_{j=1}^{l} r_j(\text{mod } n), \text{and } s_j \equiv d_j \cdot r_j^{H(t,M)}(\text{mod } n).$$

d) Broadcast s_j to all signers.

e) After every signer has received s_j, $j = 1, 2, \ldots, l$ from the others, compute s as

$$s \equiv \prod_{j=1}^{l} s_j(\text{mod } n).$$

The multisignature of a message m is $\sigma = (t, s)$. In this scheme every signer's signature is the same as Shamir's scheme.

ISBN: 978-1-61208-161-8

3) Verification Phase: To verify the multisignature $\sigma = (t, s)$ made by signers with identities $i_1, i_2, \ldots, i_l$ of m,

$$s^e \equiv (i_1 \cdot i_2 \cdots i_l) \cdot t^{H(t,m)} (\text{mod } n).$$

If this verification equation is successful, then the information has a legitimate signature.

Harn-Ren's multisignature scheme does not protect the signer's signature secret key from being exposed [6]. Anyone is able to obtain the signer's secret key d_j using broadcast data (r_j, s_j) and signature (m, σ). Moreover, if e is a small value, an attacker is able to obtain the signer's secret key d through the public information (e, m, s) .

D. The improved authentication scheme

To avoid the two mentioned loopholes in Harn-Ren multisignature scheme, a new one was proposed in [6], where the KGC keys phase and signer secret key generation phase is the same as the original scheme in §II-C.

1) Signing phase: As before, if the group of signers $i_1, i_2, \ldots, i_l$ want to jointly sign the document m, each signer j performs the same steps mentioned in §II-C, except that now the values s_j are defined as:

$$s_j \equiv d_j^t \cdot r_j^{H(t,M)} (\text{mod } n).$$

2) Verification phase: When the receiver receives multisignature message (m, σ) , the public key e of the KGC and the identities of all the signers $i_1, i_2, \ldots, i_l$ can be used to verify the validity of the signature. The verification formula is as follows:

$$s^e \equiv (i_1 \cdot i_2 \cdots i_l)^t \cdot t^{H(t,m)} (\text{mod } n).$$

If verification is successful, then the information has a legitimate signature. Otherwise, it is an illegal signature.

III. Implementation and procedures

We have developed the Harn-Ren improved multisignature scheme. We have started with Matlab, with the use of functions and toolboxes to encrypt, decrypt and sign messages with RSA cryptosystem, with real parameters, and we changed to Java to code a more efficient programm.

A. Matlab and big integers

To implement RSA cryptosystem we need to find big integers, at least 1024 or 2048 bits keys, to be sure that RSA is secure against known attacks. Trying to work with Matlab, we found a toolbox, `vpi`, to compute variable precision arithmetic operations.

First of all we started the cryptosystem implementation trying to encrypt and decrypt a short message to check Matlab possibilities. We took parameters p and q with length of about 155 digits. In this case, the calculation of public and private key is fast, and also the calculation of the encrypted message, but to get the plain text is very slow, because the modular power with Matlab is not enough efficient. This was the reason to change to Java language.

B. Java implementation

We have developed a Java a digital identity multisignature application. Although Java is object oriented, being a simple and algorithmic implementation, we have divided the program into two classes, inside the `multisignatures_identities` package. The first class is called `Identities`, this class contains the embodiment of the signature and verification. The second class name is `Hash`, and performs a hash function from a string and returns the string's digest.

`Identity` class is composed by the following attributes:

1) RSA parameters: `p, q, n, fi, d, e`.
2) Number of participants: `num`
3) Each participant has: Identity (`i_j`), Private key (`d_j`), and other data used in the multisignature (`r_j, t_j, s_j`).
4) The values of the multisignature: `s, t`.

`Identity` class contains the methods that carry out the following actions:

1) Calculation of RSA parameters:
   a) `calculate_module`: returns the module `n` when `p, q` are known.
   b) `calculate_fi_euler`: returns `fi` when `p, q` are known.
   c) `calculate_d`: returns private key `d`, when public key and `fi` are known.
2) Calculation of identities and private keys:
   a) `calculate_identities_and_private_keys`: with the number of participants `num`, the private key `d`, and the modulus `n`, the identities `i_j` and private keys for each signer `d_j` can be calculated.
3) Other estimates:
   a) `calculate_first_step`: with the number of participants `num`, the public key `e` and the modulus `n`, values `t_j` and `r_j` can be obtained.
   b) `calculate_third_step`: calculate the value `t` with some of the previously calculated parameters.
   c) `calculate_s`: obtains the value `s` with some of the previously calculated parameters.
4) Some views on screen:
   a) `initial_parameters_view`: to show the following parameters: `p, q, n, fi, d, e`.
   b) `identities_and_private_keys_view`: to show the following parameters for each participant: `i_j, d_j`.
   c) `first_step_values_view`: to show `t_j, r_j` parameters for each participant.
   d) `third_step_values_view`: to show `t, s_j` parameters for each participant.
5) Signature verification:

a) `signature_verification`: verifies the returned signature with a boolean value: true if the signature is verified or false if not.

`Hash` class is composed by a serie of attributes:

1) `md`: is `typeMessageDigest`, where the hash function is of type SHA-1.
2) `buffer`: is an array of bytes, which contains the string to calculate the hash.
3) `digest`: byte that includes the string for conversion.
4) `hast`: the string which will store the hash value.

`Hash` class includes the following methods:

1) `Hash`: is the constructor method.
2) `getHash`: will calculate the hash for a string.

To develop the proposed multisignature scheme, we have used two classes: `BigInteger` and `BigDecimal`. These classes' types have advantages over the types primitive. When big numbers are needed in Java, the best option is to use these classes. In fact, their storage limit is the same limit as the Java virtual machine memory limit.

The `BigDecimal` class is only used to generate random numbers with the `random()` method of the Math library. The `BigInteger` class was more useful to the program because of some of the methods provided by this class. The methods that were interesting for us were:

1) `multiply(BigInteger val)`: it returns the multiplication of this `BigInteger` with the input parameter.
2) `subtract(BigInteger val)`: it returns the subtraction of this `BigInteger` with the input value.
3) `mod(BigInteger m)`: it returns the value of `BigInteger` module `m`, with `m` the input value.
4) `modInverse(BigInteger m)`: it returns the inverse of this `BigInteger` module `m`.
5) `modPow(BigInteger exponent, BigInteger m)`: it returns the pow of this `BigInteger` with exponent `m`.
6) `compareTo(BigInteger val)`: it compares `BigInteger` with the parameter passed in the method and return `0` if they are equal.

C. Benefits from this developments

We have calculated the CPU time to perform an identity based multisignature and the time to verify the multisignature with the proposed Java implementation. We have used a `System` class method called `currentTimeMillis()`. This method returns the current time in milliseconds. The needed average time to multisign a document by 10 signers is 88.3ms, and 1.3ms to verify. If we take 100 signers, the time to multisign the same document is 636ms and 3.2ms to verify.

These are two benefits related to the development:

1) The time to sign and verify a document is slow.
2) The possibilities offered by java environment are good. The source code detailed in section §III-B could be added to a java card applet to get a secure environment that allows different people to multising documents.

IV. Conclusion and future work

As we presented, some identity-based identification and signature schemes have been implemented using Java, as can be shown in [9], but there is no implementation related to an identity-based multisignature scheme based on RSA. We studied the possibilities of a software like Matlab, but we recognize that it does not work properly with big integers, that are needed to encrypt, decrypt and sign messages with RSA, and to multisign messages or documents with some users, the calculations are more slowly that the case of single RSA.

We have chosen the Java programming language because of its efficiency and because we are developing some Java Card applets that enable to digital sign documents.

Acknowledgment

This work has been supported by Fundación Memoria D. Samuel Solórzano Barruso under the project FS/7-2010.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes", Advances in Cryptology (Crypto'84), vol. 196, 1984, pp. 47–53.

[2] R. Durán Díaz, F. Hernández Álvarez, L. Hernández Encinas, and A. Queiruga Dios, "A review of multisignatures based on RSA", Proceedings of The 4th International Information Security & Cryptology Conference (ISCTURKEY'10), 38–44. Ankara (Turkey), May 2010.

[3] Y.F. Chang, P.C. Chen, and T.H. Chen, "A Verifiable Identity-based RSA Multisignature Scheme for Mobile Communications," Journal of Computers, vol. 20, 3, 2009, pp. 3–8.

[4] L. Harn and J. Ren, "Efficient identity-based RSA multisignatures", Computers & Security, vol. 27, 2008, pp. 12–15.

[5] Y.F. Chang, Y.C. Lai, and M.Y. Chen, "Further Remarks on Identity-based RSA Multisignature," PRoc. Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, doi:10.1109/IIH-MSP.2009.137.

[6] F.Y. Yang, J.H. Lo, and C.M. Liao, "Improvement of an Efficient ID-Based RSA Multisignature," International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), 2010, pp. 822–826.

[7] R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Commun. ACM, vol. 21, 1978, pp. 120–126.

[8] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of applied cryptography, CRC Press, Boca Raton, FL, 1997.

[9] S.Y Tan, S.H. Heng, B.M. Goi, and J.J. Chin and S. Moon, "Java Implementation for Identity-Based Identification," International Journal of Cryptology Research, vol. 1, 1, 2009, pp. 21–32.

Connected Dominating Set Problem and its Application to Wireless Sensor Networks

Razieh Asgarnezhad
Department of Computer Engineering
Arak Branch, Islamic Azad University
Arak, Iran
raziehasgarnezhad@yahoo.com

Javad Akbari Torkestani
Department of Computer Engineering
Islamic Azad University, Arak Branch
Arak, Iran
j-akbari@iau-arak.ac.ir

Abstract— **In Wireless Sensor Networks, all nodes are energy constrained. There are no predefined and no fixed infrastructures in networks. A Connected Dominating Set can be created by different algorithms to organize nodes in a better way. A Connected Dominating Set can be shown as a backbone. A backbone is a subset of nodes that are able to perform especial tasks and serve nodes which are not in the backbone. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a Wireless Sensor Network. For example, Connected Dominating Set nodes can perform efficient routing and broadcasting in networks. This paper tries to survey and classify different Connected Dominating Set formation algorithms. We compare their performances with each other.**

Keywords- wireless sensor network; maximal independent set; connected dominating set.

I. Introduction

Wireless Sensor Networks (WSNs) have attracted recent research attention due to wide range of network communications applications they support. In WSNs, all nodes are energy constrained. They include a number of wireless nodes and they can be divided into three parts: data collection, based-station and data management center. Also, there is no fixed or predefined infrastructure in these networks. A kind of broadcasting in sensor networks is normally flooding-based, where each node retransmits the broadcasting message that it receives. But it raises energy consumption because packet retransmission is needed when interference occurs. Also, it will has broadcast storm problem. [2][10]

The extensive research performed in the past of decades in WSNs. Among the topics that have received especially attention, clustering formation and interconnection, also referred as *backbone formation*. Backbone will remove unnecessary transmission links through shutting down some of redundant nodes. Although backbone will still guarantee network connectivity in order to deliver data efficiently in a WSN. In virtual backbones based WSNs, some nodes are chosen as dominator node (backbone node) in the backbone construction process.

A backbone is a subset of nodes that are able to perform especial tasks and it serve nodes which are not in the backbone. Therefore, the backbone construction depends on the task to be carried. In WSNs, a backbone could be the set of active sensors while the rest of the sensors are sleeping. The backbone of a network is normally required to be connected, so that the backbone nodes are able to communicate to perform especial tasks. For example, to connect backbone nodes in ad hoc networks can perform efficient routing and broadcasting. A Connected Dominating Set (CDS) can be showed as a backbone. Backbones improved the routing procedure. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a WSN. [15]

The nodes in CDS are called dominator (backbone node), other nodes are called dominatee (non-backbone node). With the help of CDS, routing is easier and can adapt quickly to network topology changes. To reduce the traffic during communication, it is desirable to construct a minimum CDS (MCDS). Constructing a MCDS is proved a NP-hard problem and in recent years many algorithms of constructing an approximate MCDS have been proposed. [12][15]

We classify different CDS formation algorithms in these networks in Section 2. In Section 3, we present some examples of this classification. We compare performance of these with each other in Section 4. In Section 5, we conclude the paper.

II. Classification of CDS Formation Algorithms

We will present a new classification of CDS formation algorithms. From varied aspects, we can be classified into blew different types.

A. UDG and DGB

The CDS construction algorithms can classified into two types: Unit Disk Graph (UDG) based algorithms and Disk Graphs with Bidirectional (DGB) links. In UDG and DGB, the link between any pair of nodes is bidirectional. The nodes transmission ranges in UDG are the same, but in DGB are different. Even in UDG and DGB, MCDS is proved as a NP-hard problem. In Figure 1, we show a UDG of CDS virtual backbone. [2][13][15][17]

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

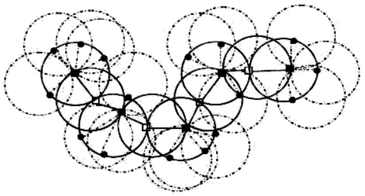


Figure 1. A sample for UDG of CDS virtual backbone [15]

B. *MIS Based and Non-MIS Based*

Independent set (IS) of a graph G is a subset of vertices so that no two vertices are adjacent in the subset. Maximal Independent set (MIS) is an IS, so that it is not a subset of any other IS. Note that in an undirected graph an MIS is also a Dominating Set (DS). The MIS based algorithms have two kinds of realization. The optimal nodes selection is based on some criterions such as node degree, rest energy of node, and node id, etc. [8] [15] [18] [19]

C. *Centralized Algorithms and Decentralized Algorithms*

Algorithms that construct a CDS can be divided into two types: centralized algorithms and decentralized algorithm. The centralized algorithms in general result in a smaller CDS with a better performance ratio than that of decentralized algorithm. The decentralized algorithms also can be divided into two types: distributed algorithms and localized algorithms. In distributed algorithms, the decision process is decentralized. But, in the localized algorithms, the decision process is not only distributed also requires only a constant number of communication rounds. Most of the distributed algorithms find a MIS then, connect this set. [3][8][9][18][19]

D. *Pruning-Based Algorithms*

Some algorithms use pruning rules to reduce the redundant nodes of backbone. In often these algorithms, all nodes of network considered to be backbone nodes for creating CDS. Then they pruned its redundant nodes to can create MCDS. [2][4][[5-6] [15] [19]

III. Some Examples of This Classification

We will present some examples of this classification and explain their approaches.

A completely localized algorithm was proposed to construct CDS in general graphs. At first, all vertices are unmarked. Then, they exchange their open neighborhood information with their one-hop neighbors. Thus, each node knows all of its 2-hop neighbors. The marking process applies the following simple rule: any vertex having two unconnected neighbors so that they are marked as a dominator. At last, the set of marked vertices form a CDS, but it had a lot of redundant nodes. There are two pruning principles so that they are provided to post-process the DS, according to the neighborhood subset coverage. Also, when two of its connected neighbors in S with higher ids can cover all of *u*'s neighbors then node *u* will be deleted from S. This pruning idea is expressed to the following general rule [11]. According to this rule, if there is *k* connected neighbors with higher ids in S so that can cover all *u*' s neighbors then, a node *u* can be removed from *S*. [19]

Guha et al. [9] proposed two CDS construction approach. The algorithm1 begins through marking all vertices white. At first, the algorithm selects the node with the maximal number of white neighbors. The selected vertex is marked black and its neighbors are marked gray. The algorithm iteratively seeks the gray nodes and their white neighbors, and it selects the gray node or the pair of nodes, whichever has the maximal number of white neighbors. The selected node or the selected pair of nodes is marked black, and also their white neighbors marked gray. The algorithm terminates, when all of the vertices are marked gray or black. All the black nodes form a CDS. This algorithm results in a CDS of size at most $2(1+H(\Delta)).|OPT|$, where *H* is the harmonic function, and *OPT* refers to a MCDS.

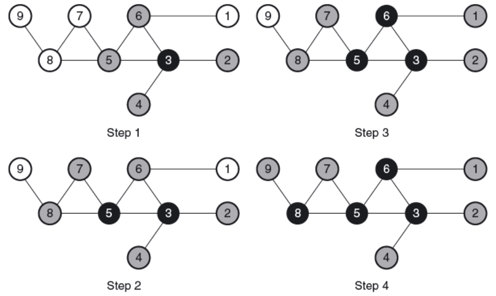


Figure 2. An example of Guha and Khuller's algorithm 1 (above) [9]

The algorithm2 also begins through coloring all nodes white. A *piece* is defined to be either a connected black component, or a white node. The algorithm includes two phases. The first phase iteratively selects a node that yield the maximum reduction of the number of pieces. A node is marked black and its white neighbors are marked gray, when it is selected. At last, the first phase terminates when no white node left. Therefore, there exists at most $|OPT|$ number of connected black components. The second phase constructs a Steiner Tree to connect all the black nodes through coloring chains of two gray and black nodes. The size of the resulting CDS formed via all black nodes is at most $(3+ln(\Delta))|OPT|$.[9]

Das et al. [8] proposed the distributed implementations of the two greedy algorithms. The first algorithm grows one node with maximum degree to be form a CDS. A node must know the degree of all nodes in the graph. Each step selects either a one- or two-edged path from the current CDS. Then the nodes in the CDS must know the number of unmarked neighbors for all nodes one and two hops from the CDS. This algorithm produces a CDS with approximation ratio of $2H(\Delta)$ in $O(|C|(\Delta+|C|))$ time, using the $O(n|C|)$ messages, where the harmonic function, *n* is the total number of vertices, and *C* represents the final CDS.

In the second algorithm, they compute a DS and then select additional nodes to connect the set. According to the

DS in the first stage, an unmarked node compares its effective degree, with the effective degrees of all its neighbors in two-hop neighborhood. The greedy algorithm adds the node with maximum effective degree to the DS. When a DS is achieved, the first stage terminates. The second stage connects the components through a distributed minimum spanning tree algorithm. At last, the nodes in the resulting spanning tree compose a CDS. This algorithm has time complexity of $O((n+|C|)\Delta)$, and message complexity of $O(n|C|+m+n\log(n))$. It have the MCDS with a ratio of $2H(\Delta)+1$, where m is the cardinality of the edge set.

Akbari et al. [2] proposed an intelligent backbone formation algorithm according to distributed learning automata (DLA). The worst case running time and message complexity of the backbone formation algorithm has a 1/(1-ε) optimal size backbone. This was why that it was shown that through a proper choice of the learning rate of the algorithm, a trade-off between the running time and message complexity of algorithm with the backbone size can be made.

In implementation, a network of the learning automata isomorphic to the UDG was used. At first, it formed through equipping each host to a learning automaton. At each stage of this approach, the learning automata randomly choose one of their actions so that a solution can be found in the CDS problem. The created CDS is evaluated via the random environment, and the action probability vectors of the learning automata are updated depending on the response received from the environment. At last, in an iterative process, the learning automata converge to a common policy and it constructs a minimum size virtual backbone for us.

This algorithm used a pruning rule to avoid choosing the same dominators. In this rule point of view, it increases the convergence speed, and also, decreases the running time of the proposed algorithm. With comparing the results of proposed algorithm with the other of the best known CDS-based backbone formation algorithms, the results show that their algorithm always outperforms the others in terms of the backbone size, and also its message overhead is only a few more than the least cost algorithm. [2]

Alzoubi et al. [3] provided two versions of an algorithm to construct the DS for a wireless network. In these algorithms, they employ the distributed leader election algorithm [6] to construct a rooted spanning tree. A labeling strategy is used to divide the nodes in the tree to be either black or gray, according to their ranks. The rank of a node is the arranged pair of its level and its id. The labeling process begins from the root node and finishes at the leaves. At first, the node with the lowest rank marks itself black and broadcasts a *DOMINATOR* message. According to the following rules, the marking process continues:

- "If the first message that a node receives is a *DOMINATOR* message, it marks itself gray and broadcasts a *DOMINATEE* message."[3]
- "If a node received *DOMINATEE* messages from all its lower rank neighbors, it marks itself black and sends a dominator message."[3]

When it reaches the leaf nodes, the marking process finishes. Just now, the set of black nodes form an MIS. At last, in the final phase the nodes connect in the MIS to form a CDS through *INVITE* and *JOIN* messages. This algorithm has time complexity of $O(n)$, and message complexity of O(n log(n)).

Rai et al. [15] proposed an algorithm for finding MCDS with using of DS. DSs are connected through using Steiner tree. The approximation algorithm includes of three stages. At first, the DS is determined through identifying the maximum degree nodes to discover the highest cover nodes. Then, it connects the nodes in the DS through a Steiner tree. At last, this tree prunes to form the MCDS. For local repair, rule k [11] is used to find the nodes so that can maintain the MCDS. This phase includes of the following steps:

- An arbitrary number say *id* is assigned to each node in the graph $G(V,E)$
- Each node is assigned white color
- The node u with maximum degree is taken from $G(V,E)$ and color as black, *i.e.* Dominator
- All the neighbor nodes of the node u are Colored
- Do step 3-4 till all the nodes in the graph $G(V, E)$ are colored either as black or gray.

Set of connectors B is found so that all the nodes in D connected. The set of D and B includes black nodes and also dark gray nodes, respectively. A node in B is connected through at most K. Set of dark gray nodes along with given D could be found via Steiner tree. Interconnecting all the nodes in D are through adding new nodes between them. Steiner nodes are nodes that are in the Steiner tree but not in set D. At last, constructed CDS will include of black and dark gray nodes.

This steps present in the following:

- Select a gray node which is connected to Maximum (K) number of black nodes, set Its color as dark gray
- Check whether the Dominating Set D
- if D gets connected stop
- else go to step *1* with $K-1$ number of Black nodes

Eventually, in the pruning phase, redundant nodes are deleted from the CDS to obtain the MCDS. These rules present in following steps:

- Select a minimum degree node u from F
- check if $N[u]$ is subset of $N[1]$ and N[2] and ...$N[n]$ where i belongs to $F-\{u\}$
- if step 2 returns *true* then remove node u and go to step 1
- Otherwise do not remove node u and go to step 1

They also proposed a local repair algorithm to take care of node's deletion.

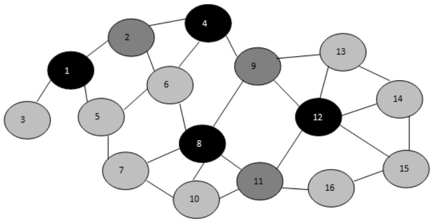


Figure 3. Show the final MCDS Backbone [15]

We have shown obtained solution of foresaid algorithm in above figure with a specific one.

Li et al. [13] proposed an algorithm for constructing CDS. They called it as Approximation Two Independent Sets based Algorithm (ATISA). The ATISA has three stages: (1) constructing a connected set (CS) (2) constructing a CDS (3) pruning the redundant dominators of CDS. ATISA constructs the CDS with the smallest size, compared with some well-known CDS construction algorithms. The message complexity of this algorithm is *O (n)*. The ATISA has two kinds of implementations: centralized and distributed. The centralized algorithm consists of three stages, which are CS construction stage, CDS construction stage, and pruning stage.

In the centralized algorithm, the initial node is selected randomly. Then the algorithm executed several rounds. When the first stage is ended, there are no black nodes generated in the network. The generated black node set is formed a CS. If a white node has black neighbors then, it will select the black neighbor with the minimum id as its dominator and it change its state into gray. If a white node only has the gray neighbors then, it will send an invite message to the gray neighbor with the minimum id and it change its state into gray. In the second stage, constructs a CDS and all the nodes are either black or gray. Finally, there is no white node left in the network. According to the third stage, if a black node with no children and if the neighbors of the black node are all adjacent to at least two black nodes, then the black node is put into connected set.

In the distributed implementation, all nodes are initialized white. After the first stage, there are white nodes, gray nodes, and black nodes. Then, in the second stage, there are black nodes, gray nodes and sometimes white nodes. White nodes can change their states into gray and also gray nodes can change their states into black. In the third stage, the redundant black nodes are deleted. [13]

Xie et al. [20] called their algorithm as Connected Dominating Set-Hierarchical Graph (CDS-HG). It is a distributed MCDS approximation algorithm. They showed that this algorithm generates smaller CDS sizes compared with the existing algorithms. Their algorithm includes of two phases. At first, in the first phase, rule1 (Essential Node Determination) is used. According to this rule, a set of dominators select for each hierarchical level so that all nodes in the next level are dominated by these dominators. A greedy strategy is used to select the dominators for creating a small initial DS. In the second phase, another rule (rule2) is used to remove the redundant dominators. This process repeated from the lowest level to the highest level of the hierarchical graph. According to The greedy strategy that created CDS is connected. Also, the size of CDS generated is at most $(logn|opt|)$, where n is the number of nodes in the network and $|opt|$ is the cardinality of a minimum DS. The computation complexity of their algorithm is $o(n^2)$. Because a centralized CDS algorithm is impractical for WSNs, Thus, they implemented a distributed algorithm based on competition. It includes of three phases: creating the initial CDS through competition and reducing the CDS size through applying rule2 on all dominators. Respectively, the computation and message complexities of their algorithm are $o(\theta^2)$ and $o(\theta)$, where θ is the maximum number of child nodes in graph. [20]

A virtual backbone was proposed for Wireless Ad-hoc Sensor Networks. According to this algorithm, the sensor network is divided into clusters. This algorithm includes of two phases. First, they clustered sensor nodes through clustering algorithm and then implemented the CDS algorithm to intra clusters. They assume all vertices are unmarked. They exchange their open neighborhood information with their one-hop neighbors. With using two pruning rules are provided to post-process the DS. If there exists a node v with higher id so that the closed neighbor set of u is a subset of the closed neighbor set of v, node u can be taken out from the CDS. [4]

Acharya et al. [1] proposed Energy-Aware Virtual Backbone Tree (EVBT) that it is a distributed algorithm for constructing a backbone in WSN. It chooses only nodes with enough energy levels as the member of the virtual backbone. Also, it introduced a concept of threshold energy level for members of virtual backbone. According to it, only nodes with energy levels above a predefined threshold are included in the EVBT. They used an undirected graph to represent a WSN. Sensor node that does not belong to the backbone is termed as *leaf node*. Every node in the network has an EVBT node. They term this EVBT node as the dominator of the corresponding leaf node. They presumed each node v knows its *N(v)*. They check two types of vertices. A tree node is a *fixed vertex* so that cannot be removed from the EVBT. It means that this vertex will be a part of the final solution. If energy level of *Non-fixed* vertices is not above threshold energy level or its removal does not disjoin the resulting sub graph, then *Non-fixed* vertices will be removed. At each step of the algorithm, at least one vertex is either fixed, or removed. It is presumed that at first, all the nodes in the network form the EVBT. At last, these non-removed and fixed vertices form the EVBT. They presumed, the sink node is leader to starts execution of algorithm.

In this algorithm, every node in the network has one virtual backbone node, which it selects as its dominator. This dominator will be parent node for that node. Any node in the network will forward its packet to its dominator. In this way the packet eventually reaches the sink node. [1]

Hussain et al. [10] constructed a CDS-based backbone to support the operation of an energy efficient network. It focused on three key ideas in their design: (1) a realistic weight matrix, (2) an asymmetric communication link between pairs of nodes, and (3) a role switching technique to prolong the lifetime of the CDS backbone. This algorithm is distributed in nature, and does not require global information. Hence, it is deterministic.

Corresponding with the weight comparison among neighbors, some suitable nodes get selected as dominators. The set of dominators is a MIS. At first, those selected dominators, in conjunction with some Connector nodes (dominator2 nodes), then on form the dominating set of the network. On the other side, nodes that are not part of the dominating set remain as dominatees, and use neighboring dominators as next hops for data communication. This algorithm presumed that all nodes know 2-hop away neighborhood information. The weight matrix used in r-CDS algorithm is: $W_i(r_i, deg_i, id_i)$. Node *i* is more suitable to be a dominator than neighboring node *j*, if any of the following is true: [10]

deg(u)- The effective node degree of node *u*
r(u)- The number of 2-hop away neighbors

- $r(i) < r(j)$
- $r(i) = r(j)$ *and* $deg(i) > deg(j)$
- $r(i) = r(j)$ *and* $deg(i) = deg(j)$ *and* $id\ (i) < id\ (j)$

According to this algorithm, sensor nodes in the r-CDS algorithm can have three different colors: white, gray and black. At first, all nodes are white. In continue, all nodes change their color to either black or gray. Black nodes form network backbone, but gray nodes remain as dominatees. In their algorithm, nodes can broadcast the following messages: *BLACK*, *GRAY* and d(u) messages. After each node knows about its two hop away neighborhood, all nodes broadcast their r values. A node u can become dominator1, if it wins in the weight comparison. Then, node *u* turns black and broadcasts a *BLACK* message in the neighborhood. If a white node v receives *BLACK* message from its neighbor *u*, so *v* becomes gray and broadcasts *GRAY* message. This *GRAY* message includes the pair (*v'* s id, *u'* s id). If a black node w receives *GRAY* message from a gray node *v* and also the id of another black node u, and if *w* and *u* are not connected yet, then *v* becomes dominator2 node to connect *u* and *w*. In that case, after receiving a *BLACK* message from a node *w*, if a gray node *u* has already received a notification so that there is a 2-hop away black neighbor *v* sent through a neighbor *x* and *v* has not been connected to *w* yet, then both *u* and *x* become dominator2 nodes to connect node *v* and node *w*. [10]

An algorithm was provided to find MCDS in UDG. It is based on the computation of convex hulls of sensor nodes. Also, it describes an algorithm to find MCDS from a CDS. This CDS is found via algorithm described in [11]. They have to do following steps: [14]

- Select a minimum degree vertex u from the CDS.
- Calculate *CH(N[u])*.
- Calculate *CH(N[i]), i ϵ N(u)*.
- Check if *CH(N[u])* is contained in *UCH (N[i])* where *i ϵ N(u)*.
- If step 2 returns true then remove vertex u and go to 1).
- Otherwise do not remove vertex *u* and go to step 1.
- Algorithm terminates when all the nodes in *C* are considered and the node remains in *C* construct the MCDS.

Stojmenovic et al. [16] According to the context of clustering and broadcasting, presented three synchronized distributed constructions of CDS. In all of these, the CDS includes of two kinds of nodes: the cluster-heads and the border-nodes. The cluster-heads form a MIS. If a node is not a cluster-head and there are at least two cluster-heads within its 2-hop neighborhood, then it is a border-node. The set of cluster-heads is extracted through three rankings such as: the id only, an ordered pair of degree and id, and an order pair of degree and location.

The selection of the cluster-heads is given via a synchronized distributed algorithm, which can be generalized to the following framework. Initially all nodes are colored white. In each stage of the synchronized distributed algorithm, all white nodes which have the lowest rank among all white neighbors are colored black. Then all white nodes adjacent to these black nodes are colored gray. Finally, the ranks of the remaining white nodes are updated. When all nodes are colored either black or gray, the algorithm ends. All black nodes form the cluster-heads. Algorithms have $O(n^2)$ message complexity and $\Omega(n)$ time complexity.

IV. Comparison of Some Algorithms

We have surveyed some well-known backbone formation algorithms in term of time and message complexity. Performance comparison of more algorithms is shown in below table. We can see that proposed algorithms in [3], [20] have the less time and massage complexity among other algorithms in this table.

Proposed algorithms in [9]-I, [9]-II result in a CDS of size at most $2(1+H\ (\Delta)).\lvert OPT\rvert$ and $(3+ln\ (\Delta)).\lvert OPT\rvert$, where *H* is the harmonic function, and *OPT* refers to a MCDS. Also [20] results in a CDS of size at most $(logn).\lvert OPT\rvert$.

TABLE I. PERFORMANCE COMPARISON

Ref.	**Performance comparison**		
	Approximation factor	*Time complexity*	*Message complexity*
[2]	-	$O(\Delta)$	$O(n\Delta^2)$
[3]	$8\ opt\ +1$	$O(n)$	$O(nlog(n))$
[8]-I	$2H(\Delta) + 1$	$O((n+\lvert C\rvert)\ \Delta)$	$O((n\lvert C\rvert +m+nlog(n))$
[8]-II	$2H(\Delta)$	$O(\lvert C\rvert\ (\Delta+\lvert C\rvert))$	$O(n\lvert C\rvert)$
[16]	n	$\Omega(n)$	$O(n^2)$
[19]	$O(n)$	$O\ (\Delta^3)$	$\Theta(m)$
[20]-I	-	$O(n^2)$	-
[20]-II	-	$O(n^2)$	$O(n)$

(*n* and *m* are the number of vertices and edges respectively, *opt* is the size of MCDS, *Δ* is the maximum degree, $|C|$ is the size of the computed CDS.)

V. Conclusion and Future Works

The CDS have proven to be an effective construct within which to solve a variety of problems that arise in WSNs. In this paper, we classified CDS formation algorithms and a few instances of these classifications. Also, we have surveyed some well-known backbone formation algorithms in term of time and message complexity. Significant attention has been paid to CDS formation algorithms yielding a large number of publications. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a WSN. The important issue that we can be reached is selection algorithm according to our use.

Acknowledgment

We would like to thank the reviewers who helped us to improve the quality of the current paper.

References

[1] T. Acharya, S. Chattopadhyay, and R. Roy, "Energy-Aware Virtual Backbone Tree for Efficient Routing in Wireless Sensor Networks," in Proc. of Int. Conf. on Networking and Services, (ICNS '07), IEEE, pp. 96-102, Athens, Greece, June 19, 2007.

[2] J. Akbari Torkestani, M. R. Meybodi, "An intelligent backbone formation algorithm for wireless ad networks based on distributed learning automata," Computer Networks 54, pp. 826–843, 2010.

[3] K. M. Alzoubi, P. J. Wan, and O. Frieder, "New Distributed Algorithm Connected Dominating Set in Wireless Ad hoc Networks," Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), IEEE, Vol. 9, pp. 297-304, 7 January 2002.

[4] R. Azarderakhsh, A. H. Jahangir, and M. Keshtgary, "A New Virtual Backbone for Wireless Ad-hoc Sensor Network with Connected Dominating Set," Third Annual Conference on Wireless On-demand Network Systems and Services (WONS), pp. 191-195, 2006.

[5] S. Butenko, X. Cheng, and Carlos A. S Oliveira, and P. M. Pardalos, "A New Heuristic For The Minimum Connected Dominating Set Problem On Ad Hoc Wireless Networks," Recent Developments in Cooperative Control and Optimization, pp. 61-73, Kluwer Academic Publishers, 2004.

[6] I. Cidon, O. Mokryn, "Propagation and Leader Election in Multihop Broadcast Environment," Proc. 12th Int. symp. Disrt. Computing, pp. 104 – 119, Greece, September 1998.

[7] X. Cheng, M. Ding, and D. Chen, "An approximation algorithm for connected dominating set in ad hoc networks," Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks (TAWN), 2004.

[8] B. Das, V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," Proc. of IEEE Conf. Communications (ICC 97), Vol. 1, p. 376-380, Montreal, Canada , June 1997.

[9] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," Algorithmica, 20(4), pp. 374-387, April 1998.

[10] S. Hussain, M. I. Shafique, and L. T. Yang, "Constructing a CDS-Based Network Backbone for Energy Efficiency in Industrial Wireless Sensor Network," In Proceedings of HPCC, pp.322-328, 2010.

[11] K. Islam, S. G. Akl, and H. Meiher, "A constant Factor Localized Algorithm for Computing Connected Dominating Sets in Wireless Sensor Networks," Proc of 14th IEEE International Conference on Parallel and Distributed Systems, (ICPADS), pp. 559-566, Melbourne, VIC, December 2008.

[12] B. Jeremy, D. Min, and T. Andrew and C. Xiuzhen, "Connected Dominating Set in Sensor Networks and MANETs," Handbook of Combinatorial Optimization, Springer, US, 2004.

[13] Z. Liu, B. Wang, and Q. Tang, "Approximation Two Independent Sets Based Connected Dominating Set Construction Algorithm for Wireless Sensor Networks," Inform. Technol. J., Vol. 9, Issue 5, pp. 864-876, 2010.

[14] G.N. Purohit, U. Sharma, "Constructing Minimum Connected Dominating Set: Algorithmic approach," International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC) Vol. 2, No. 3, September 2010.

[15] M. Rai, Sh. Verma, and Sh. Tapaswi, "A Power Aware Minimum Connected Dominating Set for Wireless Sensor Networks," Journal of networks, Vol. 4, no. 6, August 2009.

[16] I. Stojmenovic, M. Seddigh, J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 12, No. 12, pp. 14 – 25, December 2001.

[17] M.T. Thai, W. Feng, and L. Dan, and Z. Shiwei and D. Ding-Zhu, "Connected dominating sets in wireless networks with different transmission ranges," IEEE Trans. Mobile Comput. , Vol. 6, pp. 721-730, 2007.

[18] P.J. Wan, K.M. Alzoubi and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," Proc. of IEEE Conf. Computer and Communications Societies, pp. 1597-1604, New York, June 23-27, 2002.

[19] J. Wu, H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," Proc. of ACMDIALM'1999, pp. 7–14, August 1999.

[20] R. Xie, D. Qi1, and Y. Li, and J. Z. Wang, "A novel distributed MCDS approximation algorithm for wireless sensor networks," Mobile & Wireless Communications, Vol. 9, Issue 3, pp. 427–437, March 2009.

Score Manager Discovery in EigenTrust Using Virtual Magnetic Fields

Henrique Galperin, Luiz Augusto de Paula Lima Jr. and Alcides Calsavara
Pontifícia Universidade Católica do Paraná (PUCPR)
R. Imaculada Conceição 1155 - Curitiba - Paraná - Brazil
Email: {henrique.galperin,laplima,alcides}@ppgia.pucpr.br

***Abstract*—EigenTrust is a well-known distributed reputation system that uses random nodes (called Score Managers) to compute the reputation of other nodes in the network. In the original proposal, the selection of the Score Managers is made by successive hashes in a DHT. This paper proposes the usage of Virtual Magnetic Fields in replacement of DHTs in the selection of Score Managers in the EigenTrust reputation system. Virtual Magnetic Fields are self-organizing message forwarding mechanisms that are capable of delivering messages to specific nodes according to some non-functional aspects concerning the application semantics. A comparative analysis showed that the EigenTrust efficiency was improved in the proposed solution by removing the cost of the discovery of a Score Manager, and by introducing a low cost method for dissemination of information as nodes join and leave the network.**

***Keywords*-EigenTrust; reputation systems; P2P; virtual magnetic fields.**

I. Introduction

Peer accountability has long been a problem to peer-to-peer networks [1]. In order to address the issue, several distributed reputation systems have been proposed in the literature. Among them, we find EigenTrust [1], which is a reputation system that assigns to each peer a unique global trust value, based on the peer's history. It uses a distributed and secure method to compute global trust values for each peer. Those values can be used by peers to filter the interaction with other peers.

In EigenTrust, the trust value of each peer is computed by a collection of other nodes called Score Managers. The Score Managers of each peer node are randomly located by successively applying a hash function of a unique ID of the peer, such as its IP address and TCP port, resulting in a point in a DHT (Distributed Hash Table) hash space.

This paper proposes an improvement of EigenTrust replacing DHTs by Virtual Magnetic Fields for the selection of Score Managers. A Virtual Magnetic Field is a distributed self-organizing message forwarding mechanism based that allows routing of message to the most ”attractive” nodes based on an attraction force function created according to the application semantics.

This paper is organized as follows. Section II details the EigenTrust reputation system and presents the Virtual Magnetic Fields distributed message routing paradigm. After discussing the motivations for this work, Section III introduces the topology establishment and maintenance algorithm and explains how to compute the attraction strength of each node. Then, a node grouping methodology is presented whose goal is to optimize virtual magnetic planes. The proposed solution then is compared with the current solution in Section IV. Finally, future works are discussed and conclusions are drawn in Section V.

II. Related Work

In this section, the two main related works are presented, namely, the EigenTrust distributed reputation system and the the Virtual Magnetic Networks.

A. EigenTrust

EigenTrust [1] is a distributed reputation system for peer-to-peer networks. It is based on the idea that each peer in the network must have a global reputation value that reflects the experience that all nodes in the network had with it. The reputation value is first computed locally by normalizing the difference between the number of positive transactions and the number of negative transactions with a peer in order to obtain a result between 0 and 1. The global reputation of each node is computed based on the notion of transitive trust, by weighting the local trust value assigned to a node with the global reputations of it provided by other peers.

For a secure version of the reputation system, the paper proposes to use DHTs, like CAN (Content Addressable Networks) [2] and Chord [3], and to apply a hash function to the unique Id of some node x (e.g., x's IP address and TCP port) in order to map it into a point in a DHT hash space. The peer which currently covers this point will be considered the Score Manager of x. Successive hashes of the node's Id are used to determine the others Score Managers for that node. The Score Managers are responsible for consolidating the reputation of a given node.

B. Virtual Magnetic Networks

Message routing based on specific application requirements has emerged as an interesting research field due to the fast growing domain of distributed applications, especially where mobile platforms are employed. A novel message routing mechanism was introduced in [4] and [5], where the concept of *magnetic fields* is borrowed from physics to define node relationships which permit a node to attract messages, like a magnet attracts iron. Such mechanism suits applications that require messages to be delivered according to some particular non-functional requirement, that is, it can be customized according to some application needs. As an example, an application where messages carry tasks to be performed may require that every single message should be delivered to the current network node where the corresponding task can be performed in the shortest time. Naturally, that depends on node processing capacity, which changes dynamically and can be hard to manage. By employing the routing mechanism based on magnetic fields customized to attract messages to the fastest node, such application is released from the burden of managing node processing capacity information.

The magnetic-field-based routing mechanism employs an *overlay* network (represented as a directed graph), which hides the physical network, in order to model each particular application's non-functional requirement accordingly. Moreover, the mechanism permits logical node mobility, thus changing network topology, that is, an overlay network is dynamic. A *virtual magnetic network* is defined as an overlay network where each node contains a virtual magnet that attracts messages. Each virtual magnet has a *force* that represents some property associated to the target application's non-functional requirement. Thus, a node – through its virtual magnet – magnetically influences its neighbors, such that a message can be attracted from a neighbor. Magnetic influence relationships are transitive, such that a message can be attracted from an indirect neighbor, as well. The main goal is to deliver each message to the strongest node – named *pivot* – according to magnetic influence relationships, as defined by the corresponding overlay network graph, independently of the node where a message is firstly created. The mechanism assumes that each node determines its own force and, as it changes, publishes its new force to neighbors.

III. Score Manager Discovery Using Virtual Magnetic Fields

Although EigenTrust, in its original specification, uses a DHT for the selection of Score Managers, we suggest that it can be replaced by a Virtual Magnetic Network decreasing the network load in Score Managers lookup operations.

The main idea is to use one virtual magnetic plan per Score Manager. The pivot node in a plan will be the selected Score Manager. The magnetic forces of every peer are random and auditable by any other peer in the plane, to make sure that the chosen Score Manager is really random and has not been manipulated.

Since the reputation of each node can be assessed by s Score Managers (typically a global constant value) and since each Score Manager requires a particular attraction plane, we end up with $s \times N$ planes (N is the number of nodes in the network). In order to reduce this number, it is possible to group peers, allowing them to share the same set of Score Managers.

A. Motivation

The main motivation for the replacement of DHTs for Virtual Magnetic Networks is that Virtual Magnetic Networks have a proactive way of handling routing, while DHTs have a reactive way. This means that when a peer needs to know who is the Score Manager of another peer, when using a DHT, the peer will need to make a lookup in the network, contacting some nodes in the way to have the answer. This is not true when using Virtual Magnetic Networks, since the information is, in a proactive way, already known by the peer.

Even having a proactive algorithm, as proofed in the next sections, this does not cause a big overload in the network on the joining or leaving of a peer form the network, since there will only be a big number of messages exchanged when the Score Manager changes.

That fact makes the EigenTrust algorithm much more efficient in its most common operations, like the lookup of a Score Manager and the joining or leaving of a non Score Manager node, and just a few slower in the lees used operations, like the joining or leaving of a Score Manager.

B. Establishing and Maintaining the Topology of the Virtual Magnetic Networks

In order to replace DTHs by Virtual Magnetic Networks in EigenTrust, one must define how the peers inside the virtual magnetic planes are organized. Two main plane topology categories can be identified: the static topology, where there is no joining or leaving of nodes, and the dynamic topology, with nodes joining and leaving the plane at any moment.

In a static topology scenario, a good option is to construct the planes as Small Worlds [6] with a reduced number of edges, but at the same time, without increasing the average number of hops between nodes. Other techniques can be used as well, including the reproduction of the actual underlying physical topology.

When considering a dynamic topology, there is always the risk of disconnecting the virtual magnetic plane, when nodes leave the network. However, this disconnection does not prevent EigenTrust from working, but may compromise only its performance since each separate plane would have its own Score Manager. Nevertheless, all the Score Managers would still compute the same global trust value. If this situation is not avoided or minimized, the segmentation of planes tends to grow over time, resulting in an undesirable number of Score Managers. So some technique is needed to avoid plane segmentation in a dynamic network topology. Assuming that each node is capable of detecting the disconnection of a neighbor, a polling mechanism may be set up in order to minimize the probability of segmentation. A joining node x just needs to get connected to a known bootstrap peer. It is desirable, on the other hand, that x should get connected to at least some other peer chosen randomly, in order to reduce the chance of splitting the plane in case of later disconnections. If x is the new pivot, then this information is disseminated using the traditional Virtual Magnetic Networks algorithm [5].

There are two categories of nodes that can leave the network: pivots and ”regular nodes” (i.e., non pivots). If a regular node leaves the magnetic plane, then the network will be affected only if that node is part of the route from another

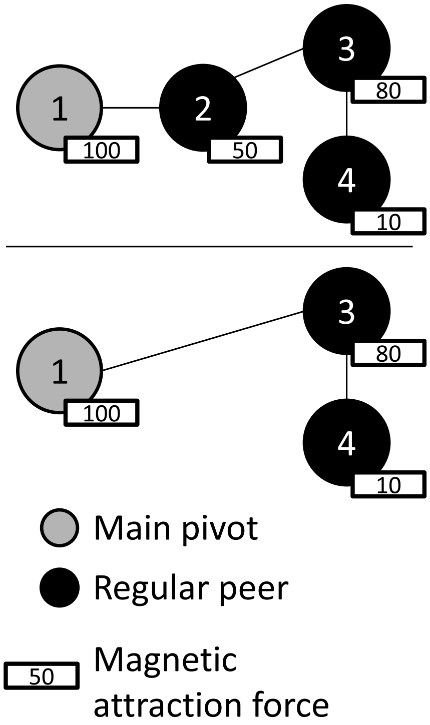


Figure 1. Leaving of a regular node that belongs to the route to the pivot.

regular node to the actual pivot. When a regular peer x detects that one of its neighbors x_n has left, that peer should check whether the missing node x_n is in its route to the pivot. If this is not the case, then nothing needs to be done. If, however, x_n is in the path to the pivot, then x will get connected directly to the pivot in order to keep the network cohesive and to restore its connection to the pivot. This is always possible because every peer knows the identity of the pivot. Since in this case the pivot has not changed, all the preexistent routes will still be valid, and no attraction force change needs to be propagated. This process is depicted in Figure 1, where node 2 leaves the plane and forces node 3 to establish a new connection with the pivot.

On the other hand, if the leaving node is the pivot, then a secondary pivot (i.e., the peer with the second largest attraction force in the plane) becomes the new pivot and its attraction strength is propagated to all nodes in the plane. If it leaves the network before the pivot itself, then the same procedure used when a regular node leaves is executed, but now a new secondary pivot has to be found.

Therefore, when a pivot leaves the network, the secondary pivot becomes the pivot, and all the neighbors of the leaving pivot check if they still have an active route to the new pivot. If they do not, they get connected directly to the new pivot. This procedure guarantees the cohesion of the plane topology. Since the pivots in the plane have changed, all attraction forces need to be propagated, starting by the nodes that lost direct connection to the previous pivot. Naturally, the virtual magnetic plane will elect a new secondary pivot and will be updated with the new pivot, using the basic force propagation algorithms of Virtual Magnetic Networks [5].

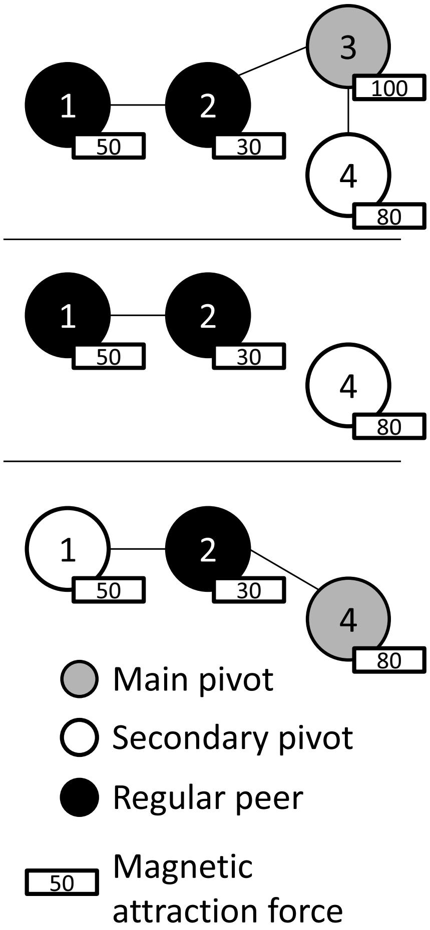


Figure 2. Leaving of the pivot.

Figure 2 depicts a scenario where the pivot (node 3) leaves the network. After that, node 2 gets connected to the secondary pivot (node 4), which will become the new pivot and then a new secondary pivot is elected (node 1, in the example).

C. Computation of Attraction Forces

Once the magnetic network topology has been established, the next step consists in defining a method to assign attraction forces to each node. Since this attraction force will be used to select the score manager for a given node (or node group), it must be unique and random within an attraction plane and different across the multiple planes.

In order to guarantee anonymity and randomness required by EigenTrust, the attraction force $F(x)$ of a given node x is defined by Equation 1.

$$F(x) = H(H(I(t)) + I(x) + k) \qquad (1)$$

where,

H is a well-known and reliable hash function, such as SHA1 [7];

$I(n)$ is the identifier of node n;

t is the node whose reputation is evaluated by the score manager;

k is a natural number used to distinguish score managers belonging to different attraction planes. For instance, if there are three score managers for each node (or group of nodes) evaluated, then k may range from 0 to 2.

Equation 1 is random due to the hash function H, guarantees anonymity by using $H(t)$ instead of $I(t)$ directly, and minimizes the probability of existing equal forces for different nodes within the same plane, since $I(n)$ is distinct for each n. Moreover, the probability of having the same score manager in different planes is minimal since k takes different sequential values.

Notice as well that all parameters of Equation 1 are known to all nodes in the magnetic plane. As a consequence, it is virtually impossible for a malicious node to fake its attraction force and to corrupt attraction force propagation data.

D. Peer Grouping

From Equation 1, one can observe that a magnetic plane can be identified by constant factors that are the same for all nodes in the plane, namely, the pair $[H(I(t)), k]$. Therefore, if we multiply the number of existing $H(I(t))$ by the number of planes (i.e., score managers) per node, we obtain the total amount of planes in the network.

Note that if there is a different $H(I(t))$ for each t (t, is a node in the network), the total number of planes can be very high, and this brings overhead to the network. In order to reduce the number of planes, multiple nodes can be grouped so that they use the same planes and the same score managers. This is done by performing the integer division of $H(I(t))$ by a constant g, for each node t. Nodes that exhibit the same result will belong to the same group, and $H(I(t))/g$ will replace $H(I(t))$ in Equation 1. For example, if there are three nodes x, y and z in a network, and $H(I(x)) = 20$, $H(I(y)) = 24$ and $H(I(z)) = 35$, then, if $g = 10$, nodes x and y will belong to the same group (since $H(I(x))/g = H(I(y))/g = 2$) and, therefore, they will share the same score managers. This strategy reduces the risk of manipulation by creating random groups.

Notice that it is important to choose the constant g according with the expected number of nodes in the network and the magnitude of H. For instance, if there are just a few nodes in the network and H produces very large numbers, it is desirable to use a very large value for g, so that to increase the probability of two nodes belong to the same group.

E. Virtual Attraction Forces Propagation

Considering that the attraction force of every peer in the magnetic pane can be calculated by any other peer, it is possible to simplify the force propagation algorithm. It is possible to propagate only the peer id, removing the attraction force from the propagation tuple, and compute the peer force locally.

IV. Comparison with the Current Solution

A comparison between magnetic field networks and DHTs as alternative approaches to implement a system to determine score managers based on EigenTrust can be made by assessing network and node resources usage in both cases. The following events must be taken into account to make a fair comparison: node join into the network, node leave from the network, and selection of score manager for an arbitrary node.

If only score manager selection is considered, it is possible to notice that the approach based on magnetic field networks presents a clear advantage, since all nodes know the pivot all the time and, consequently, no messages are needed to select the score manager, while in the approach based on DHTs, a navigation through the network is needed in order to find out which node occupies the position corresponding to the score manager, thus causing several message exchanges.

In the event of node entry and exit, there are only two cases which may cause some performance impact for the approach based on virtual magnetic networks. Firstly, if a node that joins or leaves the network is the score manager, there will be a some performance degradation, since an update regarding pivot information will be triggered. In all other cases, the consequences are not relevant. On the other hand, if for instance CAN is employed, the impact is almost always low, since in most cases, only neighbor nodes need to be notified in order to either split the existing area (when a node enters the network) or occupy a newly empty area (when a node leaves the network) if no take-over is needed. Hence, the approach based on DHTs performs better than the approach based on virtual magnetic fields in the cases where a node that enters or leaves the network is the score manager. However, the larger network, the lowest are the chances for that to happen.

Therefore, assuming that score manager join and leave happen in a much lower rate than score manager selection, the approach based on magnetic field networks will have a better overall performance than the approach based on DHTs. The costs of node entry and exit in virtual magnetic networks and in CAN are analyzed in the following sections.

A. Message Cost Analysis in Virtual Magnetic Networks

Consider the following variables:

N the number of peers in the plane;

E the number of edges (connection between two peers) in the plane, assuming that all edges are bidirectional;

Ei the number of edges created by a peer that is joining the network;

Ce the cost of creation of a new bidirectional edge;

Ea the average number of edges per peer.

The probability (P) of a peer that is joining or leaving the magnetic plane is a primary or a secondary pivot can be defined by Equation 2.

$$P = \frac{2}{N} \tag{2}$$

If a pivot changes, the number of messages needed to select a new pivot is given by Equation 3 (cost of propagation).

$$Cp = 2 \times E - N + 1 \qquad (3)$$

Based on Equation 3, it is possible to calculate the peer joining average cost (Cja) in a plane (Equation 4).

$$Cja = Ei \times Ce + P \times Cp \qquad (4)$$

Let Pr be the probability of an edge connected to a leaving node is part of a connection route to the pivot node of the node in the other side of the edge, then the average cost of a peer that leaves the network (Cla) is given by Equation 5.

$$Cla = Ea \times Pr \times Ce + P \times Cp \qquad (5)$$

Considering that the cost of creation of a new bidirectional edge (Ce) is generally low, the peer joining or leaving the network, in terms of number of messages exchanged ($Cjla$) can be approximated by Equation 6.

$$Cjla = \frac{4 \times E + 2}{N} - 2 \qquad (6)$$

Since the total number of edges (E) is function of the average number of edges per peer (Ea) and the number of peers in the plane (N) as shows the Equation 7, then $Cjla$ can be reduced to Equation 8.

$$E = \frac{Ea \times N}{2} \qquad (7)$$

$$Cjla = \frac{2}{N} + 2 \times Ea - 2 \qquad (8)$$

Notice that as N increases the peer joining and leaving average cost ($Cjla$) tends to be influenced only by the average number of edges per peer (Ea), which is independent of the network size. This result indicates that the solution proposed scales. Moreover, Ea also tends to be small, since it depends mainly on the number of edges created by a peer that is joining the network (Ei). Obviously this parameter that can be set as low as 2 and generally there is no good reason to set it to a larger value.

Notice as well that even considering that the cost of creation of a new edge on the graph (Ce) greater then zero, the final result would also be independent of the number of peers (N), which confirms the conclusion we reached.

B. Lookup Rates and Session Lengths

Regarding the lookup of score managers, the use of Virtual Magnetic Networks is clearly advantageous over traditional DHTs, since there is no need of communication, while DHTs require $O(logN)$ at best (CHORD). The information about score managers is already known by each node due to the proactive nature of the magnetic force propagation algorithm.

In the case of Virtual Magnetic Networks, all costs are transferred to the moment nodes join or leave the network. Although DHTs may show a constant cost in these situations (generally speaking, leaving nodes require updates of routing tables), notice that in $(1 - 2/N) \times 100$ percent of the cases Virtual Magnetic Network will require just a few reconnection operations. Only when the node joining or leaving the network is the pivot (primary or secondary) then propagation will be required, and the message cost will be higher. In fact, if we consider the use of CAN to implement the DHT abstraction (actually, this is the P2P (Peer-to-Peer) network suggested by the EigenTrust original specification [1]), the leaving process can be even more expensive, since it is possible that none of the neighbors of the leaving area can occupy the empty space, forcing a "take-over situation", that has no time upper bounds.

Since typical session lengths (a session is defined as join-participate-leave cycle) in P2P structured networks can be measured in hours as shown in [8] and since the probability of a pivot joins or leaves the network is small $(2/N)$, we can safely claim that the lookups outnumber by far the need of propagation due to pivot changes. This fact makes the use of Virtual Magnetic Fields more advantageous than DHTs in this context.

V. Conclusion and Future Work

We have presented an alternative to DHTs for the selection and lookup of Score Managers in the EigenTrust reputation system based on Virtual Magnetic Networks. The proposed solution provides a proactive solution for this problem, without adding significant costs as peers join and leave the network. Specifically, the method removes the need of peer lookup in order to identify the Score Managers of a particular node. As shown in Section IV, the use of Virtual Magnetic Fields in this context brings a real gain in the average network operation, providing a real gain in the average network operation.

This work is part of an ongoing research whose goal is to define a complete reputation mechanism for Virtual Magnetic Fields in an open network. Our future work therefore includes the design and implementation (possibly through simulation) of a reputation system that could weight the attraction strength informed by each node based on the reputation of that node.

References

[1] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 640–651.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2001, pp. 161–172.

[3] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2001, pp. 149–160.

[4] L. Lima and A. Calsavara, "Autonomic Application-Level Message Delivery Using Virtual Magnetic Fields," *Journal of Network and Systems Management*, vol. 18, no. 1, pp. 97–116, 2010.

[5] A. Calsavara and L. Lima Jr, "Routing Based on Message Attraction," in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2010, pp. 189–194.

[6] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[7] P. FIPS, "180-2: Secure Hash Standard," *US Department of Commerce, Technology Administration, National Institute of Standards and Technology*, 2002.

[8] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 189–202.

On-the-fly Routing and Traffic Self-Aggregation Towards Metro-Access Integration

Rodolfo Alvizu, Mónica Huerta, Ricardo Gonzalez, Roger Clotet and Laura Rodríguez
Departamento de Electrónica y Circuitos
Universidad Simón Bolívar (USB)
Sartenejas, Venezuela
alvizu22@gmail.com, mhuerta@usb.ve,
rgonzalez@usb.ve, rclotet@usb.ve,
laurarodriguez@usb.ve

Idelfonso Tafur Monroy
Department of Photonics Engineering
Technical University of Denmark (DTU)
Lyngby, Denmark
idtm@fotonik.dtu.dk

***Abstract* — The evolution and penetration of Optical Access Networks threatens to create a higher electronic bottleneck at Metro-Access interfaces caused by the ever increasing users' bandwidth demand. Since most of the newest applications are dominated by video-related traffic, there is a pressing need to relieve the electronic bottleneck between Access and Metropolitan area networks. In this work we present an enhanced version of time-wavelength access architecture with on-the-fly routing and traffic self-aggregation. The purpose of this architecture is to allow for a transparent Metro-Access integration with low latency and reduced power consumption. A WDM PON (Passive Optical Network with Wavelength Division Multiple Access) architecture was used as reference for comparison. The results obtained by the simulation model developed in OPNET Modeler show that the WDM PON will tend to produce electronic bottleneck issues, even though delays and loss rates are not very different in both architectures. These bottleneck issues can be avoided by the use of on-the-fly routing and traffic self-aggregation, which allows traffic to cope with the strict requirements of delay sensible applications. The proposed architecture is an interesting useful approach for Metro-Access integration and future access networks.**

Keywords- Future access network; on-the-fly Routing; Self-Aggregation; WDM PON; Metro-Access.

I. INTRODUCTION

The world is witnessing an explosion of high bandwidth consuming applications and an ever rising bandwidth demand by users. According to Cisco's Visual Networking Index, global IP (Internet Protocol) traffic will grow fourfold from 2010 to 2015 [1]. Globally, Internet video traffic is expected to grow from 40% of all consumer Internet traffic in 2010 to a 61% in 2015.

To successfully deliver enough bandwidth to end users, the bottleneck at access networks must be avoided. Therefore, the introduction of optical fiber to the customer sites (FTTx; stands for fiber to the x, where x stands for Home, Curb, Building, Premises, etc.) has been accepted as a solution to relieve the access bandwidth bottleneck and to cope with the ever increasing users' bandwidth demand [2][3].

The FTTH (Fiber To The Home) Council announces a continuous global growth of all fiber networks [4]. The average broadband access network speed grew 97% from 2009 to 2010 [1]. Most of this growth has been caused by deployments of PONs (Passive Optical Networks) based FTTx.

PON is a point-to-multipoint optical network, which connects an optical line terminal (OLT) at the carrier's Central Offices (COs) with several optical network units (ONUs) at customer sites. This is done through one or more 1:N optical splitters. The success of PON relies on its high bandwidth, infrastructure cost sharing and its simple maintenance and operation which results from the absence of electronic active components between the OLT and the ONUs.

As a consequence of its point to multi-point nature, the PON's upstream channel requires a multiple access technology. Today's standards and deployments of FTTx are based on Time Division Multiple access PON (TDM PON). TDM PON uses a single wavelength for downstream (CO to users) and upstream (users to CO). Upstream and downstream channels are multiplexed in a single fiber through Coarse WDM (CWDM) technology standardized according to ITU (International Telecommunication Union) G.694.2 (CWDM spectral grid). TDM PON keeps the cost of access networks down, by shared among all users the bandwidth available in a single wavelength. There are two main TDM PON standards used for mass rollouts [4][5]:

- Ethernet PON (EPON) technology: specified by the IEEE (Institute of Electrical and Electronics Engineers) as the 802.3ah standard, which is widely deployed in United States of America and parts of Europe.
- Gigabit PON (GPON) technology: specified by the ITU-T G.984 standard, which is broadly deployed in Japan and South Korea.

However, TDM PON cannot cope with future access networks' requirements regarding aggregated bandwidth, reach and power budget [5]. To solve these problems, it is widely accepted that the next evolution step for PON architectures is the introduction of wavelength division multiple access (WDM PON) [2][3][5]. The WDM PON approach assigns an individual wavelength to each ONU. This strategy allows the use of higher bandwidth for each ONU, a longer reach, better scalability towards higher users'

ISBN: 978-1-61208-161-8

concentration, and additionally provides transparent bit rate channels ONU-CO [3][5].

The continuous users' bandwidth demand growth is leading to 100 giga bits per second optical access systems [6]. This scenario implies that COs, supporting higher concentration of customers (with split ratio extended beyond 1:64), will have to aggregate and disaggregate traffic in volumes reaching tera bits per second. Thus, there will be a much higher congestion when managing the increasing bandwidth demand at the Metro-Access interface, and future applications' higher requirements on bandwidth guarantee and low delay application flow.

In this paper, we present an accurate performance assessment of an optical access network architecture that was originally proposed in [8]. This architecture introduces a time-wavelength (t-λ) routing for an on-the-fly (passively) routed, self-aggregating Metro-Access interface. The t-λ routing architecture, based on nonuniform traffic distribution in access networks, introduces lightpaths toward the most requested destinations, in order to relieve the electronic bottleneck and to simplify the Metro-Access interface.

Wieckowski et al. [8] have explained the advantages of the t-λ routing architecture over WDM PON. However, there is a situation in their simulation model that produces delay variations and packet reordering problems. The present enhanced version of t-λ routing architecture avoids those problems, thus allowing traffic to cope with the strict requirements of delay sensible traffic.

The simulation results obtained show that by using this enhanced t-λ routing architecture, the COs become congestion-free, a reduction of the network's power consumption is achieved, and the network is able to address different traffic's requirements. Therefore, the present work proves that the use of on-the-fly routing and traffic self aggregation is a very attractive approach for Metro-Access integration and future access networks.

The paper is organized as follows: Section 2 describes the Metro-Access interface problem and the t-λ routing optical access architecture. Section 3 presents related work on the t-λ routing architecture. Section 4 describes the simulation model developed and presents some simulation results. Section 5 describes concludes the paper.

II. Metro-Access Interface and Network Architecture

The t-λ routing optical access architecture is based on the Metropolitan and Access networks traffic profile.

A. Trafic profile of Metropolitan and Access Networks

Typical carrier's network architectures have been deployed with one or at least a few interfacing nodes (gateways) between different network segments (Access-Metropolitan-Core). Therefore, there is more traffic that goes along different network segments through interfacing nodes (remote traffic) than traffic that goes along the same network segment nodes (local traffic) [7][9].

It has been observed that in a multiple gateway scenario traffic demands at IP routing nodes are not evenly distributed among all destinations. About four to five major destinations comprise the 80-95% of the outgoing traffic in the routers, and 50-70% of the traffic goes to one major destination [7]. This behavior can lead to insufficient capacity and can cause traffic bottlenecks at some points of the network.

General routing tasks rely on electronic processing, thus Optical-Electrical-Optical (O-E-O) conversions are needed. Each O-E-O conversion implies a number of complex operations (signal amplification, detection, error correction, buffering, frame searching, extraction from buffer, packet assembly and signal emission) resulting in high complexity and power and time consumption.

B. Time-Wavelength (t-λ) Routing Optical Access network Architecture

The t-λ routing optical access architecture exploits the traffic profile of access networks to introduce time-wavelength routing for a passive and self-aggregating Metro-Access interface [8]. The goal of the architecture is to use the nonuniform traffic distribution of access networks to select a portion of the traffic and forward it through passive channels to perform the on-the-fly routing (passive optical routing). By taking advantage of the reduced (electronic) processing achieved with on-the-fly routing, the electronic bottleneck can be relieved.

The selection of the traffic portion that will be routed on-the-fly relies on the analysis and evaluation of a multiple gateway traffic-distribution. In this distribution up to 70% of the traffic in the access networks is destined to a Metro-Access gateway (interfacing node), the so called major destination [7]. This implies that:

- Major destination traffic is sent through passive channels using on-the-fly routing at the COs.
- Minor destination traffic (local network traffic) is sent through electronically routed channels using common stop-and-forward policies.

The underlying idea of the t-λ routing approach is presented in Figure 1 using a PON to connect N=3 ONUs. The architecture arranges ONUs in groups that are equal to the number of wavelength in the t-λ frame (3 wavelengths in the case depicted in Figure 1). For the upstream channel the CO makes a time-wavelength assignment and sends a frame of interleaving Continuous Wave (CW) seed light to the ONUs.

Each ONU sorts packets at the customer sites on the basis of their destination. Colorless reflective modulators are used to transmit data by modulating the CW according to the predefined time-wavelength assignment [10].

A collision free optical self aggregation will be achieved by assuring that the upstream signals arrive at the remote node (splitting and combining point) properly adjusted in time (see Figure 1). Thus, each wavelength at the CO will contain data from several ONUs. The self aggregated traffic arrives at the CO, where passive channels are routed on-the-fly (based on the pre-defined wavelength assignment), while other packet traffic is sent to local processing units for destination inspection, routing and forwarding (electronically routed channels). As shown in Figure 1, the CO assigns λ1 as passive channel. Hence λ1 is self-aggregated and on-the-fly routed towards the major destination.

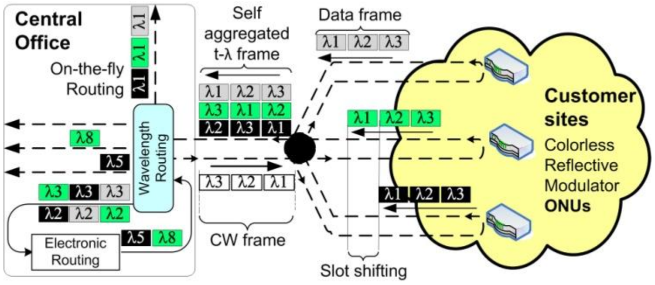


Figure 1. Underlying idea of the time-wavelength assignment and Central Office (CO) schematic diagram, for time-wavelength routed architecture to provide on-the-fly routing to Major destinations. CW: Continuous Wave.

III. Related Work

The fast penetration and the evolution of optical access networks will produce higher congestion at COs and Metro-Access interfacing nodes. Consequently, there will be a large pressure to deal with the increasing Quality of Service (QoS) demand of future applications requirements. Thus, future-proof, cost and energy efficient Metro-Access interfaces have been extensively investigated [8][10][11].

In a former performance evaluation the authors have shown the advantages of t-λ routing architecture over a conventional WDM PON. It was therefore proposed as solution for a transparent and self aggregating Metro-Access integration [8].

Nevertheless in the former evaluation the architectures were assessed by means of dedicated discrete step-based simulation models, with a Poisson process for traffic arrivals and packet Loss Rate (*LR*) as the performance metric [8]. Furthermore, an improvement to the simulation model developed in the prior evaluation can be attempted because the excess traffic in passive channels (on-the-fly routed) originally was distributed to electronically routed channels.

The excess traffic in passive channels constitutes major traffic (traffic destined to a major destination) arriving at ONUs that produces overflows of buffers associated with the passive channels (passive buffers). Therefore, a portion of the major traffic goes through on-the-fly routed channels and, in the presence of overflows, another portion (the excess traffic) goes through electronically routed channels towards the major destination. This approach allows to use passive channels resources to the maximum and to distribute major excess traffic to the electronically routed channels.

LR was the only performance metric used in the former evaluation, and advantages of the t-λ routing architecture over WDM PON have been shown [8]. But users are demanding applications with requirements beyond bandwidth and loss rate. According to Cisco's Visual Networking Index, globally Internet video traffic will grow from 40% of all consumer Internet traffic in 2010 to 61% in 2015 [1]. Inelastic traffic applications especially video-related traffic has been flooding the networks, and future optical access networks must address the requirements of low delay and jitter.

The packets will arrive at major destination with high delay variations by distributing the excess traffic in passive channels through electronically routed channels. This behavior will lead to a constant packet reordering at the major destination. Packet reordering can be compensated with buffering at the price of an additional delay. In future optical access networks (managing traffic volumes of tera bits per second) the buffer size for packet reordering could be prohibitive. We analyze two types of Internet traffic for discussing the traffic behavior and the consequences of this approach:

1) Inelastic traffic (voice and video-related traffic): The packet reordering compensation adds additional delay. Thus, inelastic traffic will be most likely to be discarded because, even when reaching the destination, it will not fulfill the low delay and jitter requirements. Additionally, this traffic (most likely to be discarded) will compete inside the t-λ routing architecture for resources with minor destination's traffic.

2) Elastic traffic (web, mail): TCP (Transport Control Protocol), the standard Internet transport protocol for non delay sensible traffic, has been proven to be packet reordering-sensible. TCP produces unnecessary traffic retransmission under constant packet reordering, and can lead to bursty traffic behavior, thus increasing the network's overload [12].

IV. Performance Assesment

In the present work, the enhanced t-λ routing architecture and a conventional WDM PON architecture (as reference) have been evaluated by means of simulation models developed in OPNET Modeler, an event-based state-of-the-art network modeling and simulation tool [13].

A. Simulation Model

The simulation models are composed of ONU nodes and CO nodes, which are connected by point-to-point links with several channels (i.e., wavelengths). There is also a Paths Computation Entity node used to create the routing tables of the nodes and establish the on-the-fly routed paths, based on a variation of the Dijkstra algorithm using the DJK OPNET Modeler packet [13]. The Metro-Access Interfacing Node is a CO node with different attributes configuration.

Each ONU connected to a CO has a dedicated wavelength in the WDM PON models. All the wavelengths are terminated at COs to apply electronic routing.

The t-λ frame in the t-λ routing architecture is composed by the same number of wavelengths as ONUs managed by the CO. Thus each ONU time shares the same number of wavelengths as there are ONUs connected to the same CO. The CO establishes on-the-fly routed paths from each ONU towards the major destination. Hence, those wavelengths associated with the major destination are passively routed at the COs.

The models built in the event-based simulation tool OPNET Modeler enable a more accurate evaluation. The main performance metrics of the model are:

- Traffic Loss Rate (*LR*): relation between bits loss and bits sent by the ONUs.
- End-to-End Delay (*EED*): time difference between the time instant when a packet's first bit arrives at

ONU and the time instant when the packet´s last bit is received at destination.

The Offered Load has been used to assess the architecture under various loads. This translates into different degrees of congestion.

It is well known that Internet traffic presents self-similar behavior. Thus the ONU nodes generate the network traffic with self-similar arrivals processes, using a variation of the OPNET raw packet generator [13].

Because the t-λ routing architecture has been proposed as a solution for a transparent and self aggregating Metro-Access integration, it must consider that inelastic traffic will flood future networks. Therefore, in the developed t-λ routing architecture model (enhanced version) the excess traffic in passive channels is dropped at the ONUs. Thus the delay variations and the packet reordering problems are avoided (see Section 3).

By dropping the excess traffic at ONUs the electronic routing tasks are reduced, avoiding the electronic bottleneck at COs (i.e., it moves the network bottleneck for major traffic from the COs to the ONUs). The fact is that, if some packets should be discarded by a network bottleneck, it is better that this happens as soon as possible. And this is precisely what the proposed enhanced t-λ routing architecture does.

B. Performance Assessment Escenario

Figure 2 presents the simple network topology used in the simulation experiments. The topology was selected in order to get the possibility to establish some comparison with the former evaluation [8]. It consists of five COs with four ONUs connect to each CO and a Metro-Access Interfacing Node (MN) as major destination. The COs were connected to each other in a ring arrangement. Just one CO is connected to the MN.

The performance assessment scenario's set up was as follows. Transmission Rate R = 125 mega bits per second (one magnitude order below EPON standard rates). The processing rate of the nodes was set to be on line with the transmission rate. Buffer sizes were assigned to limit the maximum buffer delay at 1 milli seconds in relation with the transmission rate; based on design considerations assumed in [14].

The applied traffic model has self-similar arrivals processes with Hurst parameter $H \sim 0{,}74$; based on empirical traffic evaluations [15]. The traffic distribution was the same as used in the former evaluation and was taken from a multiple gateway traffic assessment, where 70% of the offered load (A) is destined to the major destination (i.e., the MN) and the rest is equally distributed among the minor destinations (i.e., the five COs)[7][8].

The t-λ frame period was set to 125 micro seconds in the t-λ routing architecture; compatible with the Synchronous Digital Hierarchy (SDH).

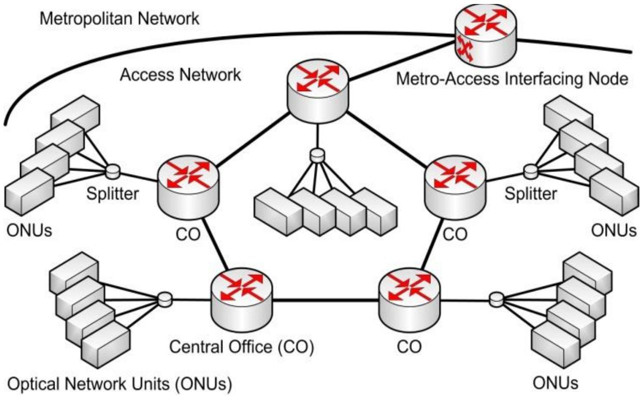


Figure 2. Simple Access network topology used for the performance evaluation, based on a ring interconection of COs. The Metro-Access interfacing Node (MN) represents the major destination (for the performance assessment up to 70% of traffic was destined to MN).

Each frame is composed of four t-λ slots (same number as ONUs connected to each CO), which were assigned in such a way that wavelengths were associated with the MN as the major destination and the COs as minor destinations.

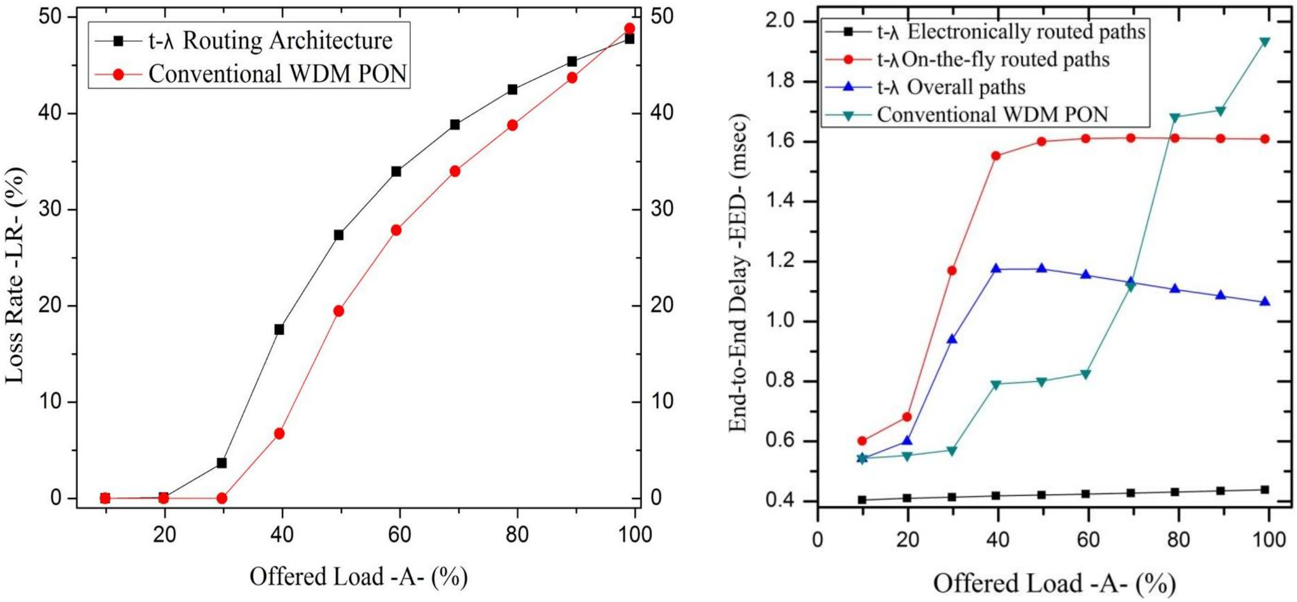
In conventional WDM PON each ONU had its own dedicated wavelength, which is terminated at the CO, i.e., no passive channels are provided.

C. Simulation Results and analysis

Performance assessment was carried out using a worst case electronic bottleneck scenario; i.e., up to 70% of the offered load (A) is destined to the major destination (MN). As there are four ONUs per each CO (see Figure 2), the t-λ frame is composed of four time shared wavelengths. Only one wavelength is passively routed towards major destination per each CO (i.e., one on-the-fly routed path established from CO to MN). In this way each ONU perceives up to 25% of the transmission rate to send traffic into the passive channel, producing fast overload of the passive channels.

Figure 3 shows the simulation results for *LR* (Loss Rate) and *EED* (End-to-End Delay) vs. *A* (Offered Load) for the enhanced t-λ routing architecture and the conventional WDM PON as reference. As can be observed in Figure 3(a), the conventional WDM PON has a superior performance, based on *LR*, because of the expected fast overload of passive channels in the enhanced t-λ routing architecture. However the *LR* in WDM PON tends to worsen when *A* increases (higher degree of congestion in the network) as a consequence of the COs' electronic bottleneck.

Figure 3(b) presents the *EED* experienced by the enhanced t-λ routing architecture, as based on three curves: electronically routed paths, on-the-fly routed paths, and overall paths. Only one WDM PON *EED* curve is shown, because all packets are electronically routed in WDM PON.



(a) (b)

Figure 3. Simulations Results for 70% of *A* (Offered Load) destined to Metro Node (MN).

(a) Loss Rate (*LR*) vs Offered Load (*A*); (b) End-to-End Delay (*EED*) vs. *A*

Figure 3(b) shows that the electronically routed paths of enhanced t-λ routing architecture presents the lowest *EED* ∀ *A*, because minor destination traffic (local traffic) perceives congestion free COs. The on-the-fly routed paths *EED* curve suggests that the ONUs passive buffers were overloaded for $A \geq 40\%$. In Figure 3(b) the enhanced t-λ overall *EED* curve indicates that when the passive buffers were overloaded ($A \geq 40\%$) there is an increasing portion of major traffic that is lost, as is clearly showed in Figure 3(a).

The WDM PON *LR* and *EED* curves depict the WDM electronic bottleneck problem. Even though we have moved the bottleneck from COs to the ONUs in the enhanced t-λ architecture, producing a fast overload of the passive channels; the WDM PON performance tends to be worse when the network is highly loaded ($A \geq 70\%$). Figure 3(b) shows that by using WDM PON the COs tend to get congested when the network load is increased. For $A \geq 70\%$ the WDM PON *EED* becomes worse than the *EED* experienced by the enhanced t-λ architecture.

V. Conclusion and Future Work

In a future scenario with higher users' bandwidth demand for video-related traffic (61% of all customer traffic for 2015) and faster networks, the access network must successfully deliver the demanded bandwidth and cope with the strict traffic requirements on low delay and jitter.

A former t-λ routing architecture model feature was found which produces constant packet reordering and performance problems. To deal with this issue we have proposed an enhanced version of the t-λ routing architecture which effectively avoids the reordering packet problems. The proposed scheme was evaluated against a traditional WDM PON architecture using the OPNET Modeler tool.

Even though EED on WDM PON and t-λ routing are not very different, our simulation results showed that the WDM PON leads to congestion at COs in presence of nonuniform access traffic distribution, as a consequence of the electronic bottleneck. In spite of the passive channel's fast overload at ONUs, the use of the proposed enhanced t-λ routing architecture, allows the COs to remain congestion free, there is reduction of the network´s power consumption and the network can more efficiently support different traffic requirements.

The introduction of on-the-fly routed passive channels based on the nonuniform access traffic distribution could represent an interesting useful solution for transparent and low latency Metro-Access integration.

It would be convenient to conduct some additional experiments introducing inelastic and elastic traffic differentiation. Using traffic differentiation can assure that only inelastic traffic will be sent by the passively routed channels, whereas elastic traffic could be sent toward electronically routed channels.

In a Metro-Access integration scenario each CO must manage much more than 64 ONUs. However the t-λ routing architecture presents a limitation in the number of wavelengths that each ONU will manage (the t-λ frame is composed of the same number of wavelengths as ONUs managed by the CO). In consequence, the scalability on the number of ONUs managed by each CO is limited.

Based on our simulation results, we propose to combine WDM PON with the use of on-the fly routing and self aggregation for traffic going toward the major destination (i.e., Metro-Access Interface). This combination is possible by means of hybrid WDM with Sub Carrier Multiple Access (SCMA) or with Optical Code Division Multiple Access

(OCDMA). Such a combination would provide transparent ONU-CO connections and transparent Metro-Access on-the-fly routed paths (releasing electronic bottleneck) without restrictions to increase the number of ONUs per CO.

ACKNOWLEDGMENT

Part of this work has been supported by the Applied Networks and Telematics Group at Simón Bolivar University (GRETA-USB), Venezuela, the Metro Access and Short Range Systems Group at Technical University of Denmark. The author would like to thank the Venezuelan “Fondo de Investigación y Desarrollo de las Telecomunicaciones” – FIDETEL (Telecommunications’ Research and Development Fund) for funding this research.

REFERENCES

[1] Cisco's Visual Networking Index, Forecast 2010-2015. 1.06.2011.

[2] L. G. Kazovsky, W. -T. Shaw, D. Gutierrez, N. Cheng, and S. -W. Wong, "Next-Generation Optical Access Networks," J. Lightwave Technol., vol. 25, pp. 3428-3442, 2007.

[3] C. Gee-Kung, et al., "Key Technologies of WDM-PON for Future Converged Optical Broadband Access Networks [Invited]," Optical Communications and Networking, IEEE/OSA Journal of, vol. 1, pp. C35-C50, 2009.

[4] Fiber-to-the-home Council http://www.ftthcouncil.org. 12.09.2011

[5] K. Grobe and J. P. Elbers, "PON in adolescence: from TDMA to WDM-PON," Communications Magazine, IEEE, vol. 46, pp. 26-34, 2008.

[6] J. I. Kani, et al., "Next-generation PON-part I: Technology roadmap and general requirements," Communications Magazine, IEEE, vol. 47, pp. 43-49, 2009.

[7] P.-L. Tsai and C.-L. Lei, "Analysis and evaluation of a multiple gateway traffic-distribution scheme for gateway clusters," Computer Communications, vol. 29, pp. 3170-3181, 2006.

[8] M. Wieckowski, A. Osadchiy, J. Turkiewicz, and I. Monroy "Performance assessment of flexible time-wavelength routing for a self-aggregating transparent Metro-access interface," in Optical Communication, 2009. ECOC '09. 35th European Conference on, 2009, pp. 1-2.

[9] E. Van Breusegern, et al., "Overspill routing in optical networks: a true hybrid optical network design," Selected Areas in Communications, IEEE Journal on, vol. 24, pp. 13-25, 2006.

[10] A. V. Osadchiy, J. Bevensee Jensen, P. Jeppesen, and I. Tafur Monroy, "Colorless receiver enabling crossconnect based metro-access interfacing nodes for optically labelled DQPSK payload signals," in IEEE Lasers and Electro-Optics Society, 2008. LEOS 2008. 21st Annual Meeting of the, 2008, pp. 612-613.

[11] J. Segarra, V. Sales, and J. Prat, "An All-Optical Access-Metro Interface for Hybrid WDM/TDM PON Based on OBS," J. Lightwave Technol., vol. 25, pp. 1002-1016, 2007.

[12] L. Ka-Cheong, V. O. K. Li, and Y. Daiqin, "An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges," Parallel and Distributed Systems, IEEE Transactions on, vol. 18, pp. 522-535, 2007.

[13] OPNET Modeler. OPNET Technologies Inc. http://www.opnet.com/. 12.09.2011

[14] B. Skubic, C. Jiajia, J. Ahmed, L. Wosinska, and B. Mukherjee, "A comparison of dynamic bandwidth allocation for EPON, GPON, and next-generation TDM PON," Communications Magazine, IEEE, vol. 47, pp. S40-S48, 2009.

[15] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," Networking, IEEE/ACM Transactions on, vol. 2, pp. 1-15, 1994.

ISBN: 978-1-61208-161-8

Implementation of Data Distribution Service Listeners on Top of FlexRay Driver

Rim Bouhouch, Wafa Najjar, Houda Jaouani, Salem Hasnaoui
SYSCOM Laboratory
National Engineering School of Tunis
Tunis, Tunisia
{rim.bouhouch@yahoo.fr, wafa_najjar@yahoo.fr, jouani_houda@yahoo.fr, salem.hasnaoui@enit.rnu.tn}

***Abstract*-In this paper, we present a way to use Data Distribution Service Listeners implemented in the C language over FlexRay driver under µC-OSII. Our method is based on implementing all the DDS Listeners as callback operations and storing each kind of Listeners in a vector of linked list. The main goal is to create an interaction between the FlexRay Driver's Read and Write operations and the DDS Listeners to define a default communication behavior for the Interrupt Service Routine. Since every real-time network works as basis software for regular middleware such as AUTOSAR, we propose the use of the real-time middleware DDS, which has a wide range of qualities of service. Specifically, we explain in this paper our approach to implement DDS on top of FlexRay driver and its related state manager using the DDS Listeners.**

Keywords-DDS; FlexRay; Listeners; callback functions; ISR; µC-OSII.

I. Introduction

Real-Time Networks are the main field of communication systems studies since all new generations of applications not only have distribution requirements, but also are subject to deadlines. The uses of these networks vary from military to vehicular networking.

This kind of networks needs a driver to specify and control its functionality according to the normalized protocol and releases. The aim of the driver is to manage the communication system in the low communication layers regardless of the application layer. On the other hand, the OMG (Object Management Group) Data Distribution Service (DDS) provides a real-time Middleware that ensures the interaction between the physical layer and the application layer providing a communication pattern. In this paper, we will see how DDS Entities Listeners are implemented in the C language, and how to use them with the FlexRay driver according to the occurred interrupt service routine. Listeners allow communication between the middleware entities, FlexRay driver's tasks and real-time applications. We chose the C language because it is not only an embeddable language, but it is also compatible with FlexRay driver API description language under µC-OSII, the real-time OS (operating system). Also, the C language allows us to easily create the blocks Simulink of the communication model.

II. DDS Overview

DDS is a real time middleware specified by the OMG based on Subscriber/Publisher communication model [1].

The OMG DDS specification defines a data-centric communication standard for a wide variety of computing environments, ranging from small networked embedded systems up to large-scale information backbones. DDS provides a scalable, platform-independent, and location-independent middleware infrastructure to connect information producers (Publishers) with consumers (Subscribers). DDS also supports many quality-of-services (QoS) properties, such as asynchronous, loosely-coupled, time-sensitive and reliable data distribution at multiple layers (e.g., middleware, operating system, and network).

At the core of DDS is the Data-Centric Publish-Subscribe (DCPS) model, which defines standard interfaces that enable applications running on heterogeneous platforms to write/read data to/from a global data space in a net-centric system. Applications can use this global data space to share information with other applications by declaring their intent to publish data that are categorized into one or more topics of interest to participants. Similarly, applications can use this data space to access topics of interest by declaring their intent to become subscribers.

The underlying DCPS middleware propagates data samples written by publishers into the global data space, where it is disseminated to interested subscribers. The DCPS model decouples the declaration of information access intent from the information access, thereby enabling the DDS middleware to support and optimize QoS-enabled communication.

The following DDS entities (also shown in Fig. 1) are involved in creating and using a DCPS-based application:

- **Domain** – DDS applications send and receive data within a domain, which provides a virtual communication environment for participants having the same domain id. This environment also isolates participants associated with different domains, i.e., only participants within the same domain can communicate, which is useful for isolating and optimizing communication within a community that shares common interests.
- **Domain Participant** – A domain participant is an entity that represents a DDS application's participation in a domain. It serves as factory, container, and manager for the DDS entities described below.
- **Data Writer and Publisher** – Applications use data writers to publish data values to the global data space of a domain. A publisher is created by a domain participant and used as a factory to create and manage a group of data writers that publish their data in the same logical partition within the global data space. Data writers and publishers have related QoS policies that drive their behavior as DDS entities.

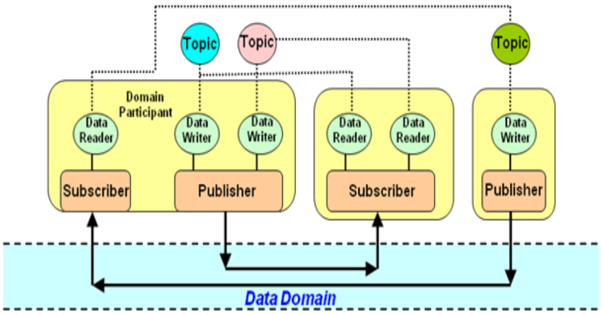


Figure 1. DDS Architecture

- **Subscriber and Data Reader** – Applications use data readers to receive data. A subscriber is created by a domain participant and used as a factory to create and manage data readers. A data reader can obtain its subscribed data via two approaches, as shown in Fig. 2: (1) listener-based, which provides an asynchronous mechanism to obtain data via callbacks in a separate thread that does not block the main application and (2) waitset-based, which provides a synchronous mechanism that blocks the application until a designated condition is met.

- **Topic** – A topic connects a data writer with a data reader, i.e., communication does not occur unless the topic published by a data writer matches a topic subscribed to by a data reader. Communication via topics is anonymous and transparent, i.e., publishers and subscribers need not be concerned with how topics are created or who is writing /-reading them since the DDS DCPS middleware manages these issues [3].

III. State of Art

The OMG DDS is an API specification and an interoperability protocol that defines a data-centric publish-subscribe architecture. Since the DDS creation, the OMG has identified several implementations including RTI [6] (Real Time Innovations) Inc, which has developed the Java DDS, the PrismTech OpenSplice DDS [2] [7] that provides an academic and commercial API in the C and the C++ languages and a multitude of others companies [8] that provide a minimum profile of DDS. All these implementations are either too voluminous to be embedded or operate over a CORBA (Common Object Request Broker Architecture) platform [9]. Our approach is different, in that we associate DDS with a real-time network like CAN [10] or FlexRay.

The goal is to take advantage of the wide range of QoS available in the DDS specification and associate it with a real-time network to improve its performances, considering that DDS provides real-time QoS like Deadline or Latency Budget. But, the actual challenge of this association is how to ensure the interaction between the middleware and the real-time network, our approach is to integrate some DDS components and functionality into the network driver. The purpose is to guarantee that the driver Read and Write operations are taken in charge by the DDS Reader and Writer.

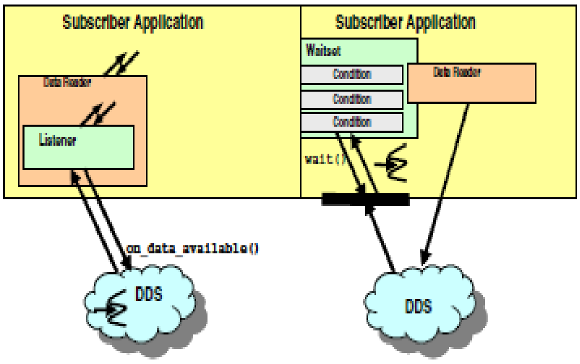


Figure 2. Listeners and wait-set Notifications[2]

Listeners and conditions (in conjunction with wait-sets) are two alternative mechanisms provided by DDS allowing the application to be made aware of changes in the communication status.

Since the condition mechanism involves the creation of a set of StatusCondition, ReadCondition and QueryCondition, we have chosen to use the Listeners mechanism to trigger the driver operations.

IV. DDS IDL to C Mapping Rules

In this section, we give some summarized rules to get DDS-DCPS API in the C language from its IDL specification. In fact, the OMG DDS specification is given as an idl (Interface Description Language) file, which is organized into Modules, Interfaces and operations. The Interface Description Language can be mapped into several programming languages including the C according to specific mapping rules. These rules can be applied to DDS-DCPS idl specification in order to get the correspondent C-API. The results obtained by this method have modified the naming of the DDS constants, types, entities and operations; thus each of these elements name is prefixed by the module name DDS_.

Example: the interface Entity in the idl description is mapped into DDS_Entity in the C API.

V. DDS Listeners Pattern

The Listener provides a generic mechanism for the Data Distribution Service to notify the application of relevant asynchronous status change events, such as a missed deadline, violation of a DDS_QosPolicy setting... The Listener is related to changes in communication status.

Each DDS_Entity can be associated with a listener, but the implementation of these Interfaces must be done by the application. Therefore, the following Listeners are available:

- *DDS_DomainParticipantListener*
- *DDS_TopicListener*
- *DDS_PublisherListener*
- *DDS_DataWriterListener*
- *DDS_SubscriberListener*
- *DDS_DataReaderListener*

All the operations associated with each Listener must be implemented, but it is up to the application whether an operation is empty or contains some functionality [2].

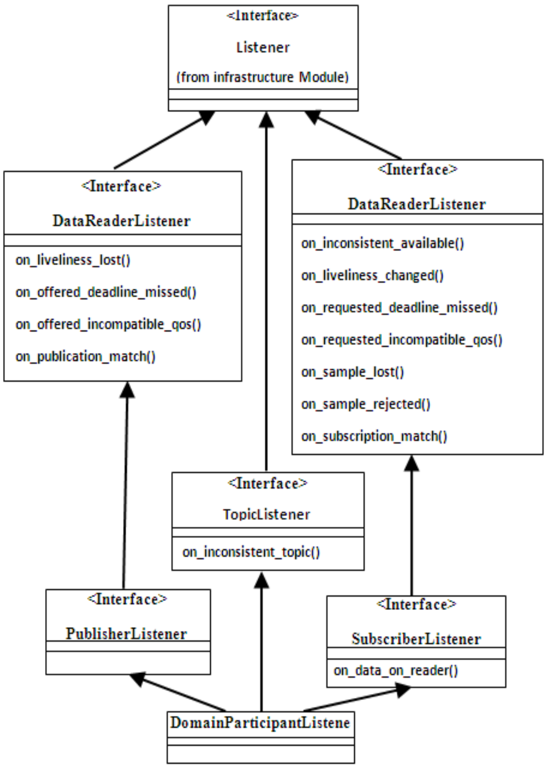


Figure 3. DDS Listeners Inheritance [4]

The structure DDS_<Entity>Listener represents the implementation of the Listener for an <Entity>. Since a Listener is implemented as a structure of pointers, the application must allocate this structure and initialize these pointers; all the function pointer attributes within the structure must be assigned to a function.

The entities listeners can inherit from each other those inheritances, shown in Fig. 3, are conformed to the classes inheritances presented in the DDS specification PIM model.

Example: *the DDS_DataReaderListener structure of Pointers:*

```
#include <dds_dcps.h>
Typedef struct DDS_DataReaderListener *
DDS_DataReaderListener;
struct DDS_DataReaderListener
{
void *listener_data;
DDS_DataReaderListener_RequestedDeadlineMissedList
ener on_requested_deadline_missed;
DDS_DataReaderListener_RequestedIncompatibleQosLis
tener on_requested_incompatible_qos;
DDS_DataReaderListener_SampleRejectedListener
on_sample_rejected;
DDS_DataReaderListener_LivelinessChangedListener
on_liveliness_changed;
DDS_DataReaderListener_DataAvailableListener
on_data_available;
DDS_DataReaderListener_SubscriptionMatchListener
on_subscription_match;
DDS_DataReaderListener_SampleLostListener
on_sample_lost;
};
```

VI. USE OF CALL BACK FUNCTION TO IMPLEMENT LISTENERS

A call back function represents the storage of the address of a sequence of instructions that the execution will trigger later on a precise event.

In fact, it involves handing over to a third routine, passing to it as an argument the address of one of our duties, so it can then call it when it needs it. In the implementation, we perform the routine that we want, but what is certain is that to save the address and call a function, it takes a function pointer that is used to perform an action, which is not known at the time of writing the code (system, library…).

All the entities listeners in DDS-DCPS are implemented as callback functions so that the notification process can be event triggered. Each listener is a structure of pointers that refers to a specific communication status change. Modifying a pointer, results in the execution of the associated function representing the default behavior of DDS regarding the happening event, as shown in Fig. 4.

The Listener is invoked on the changes of communication statuses. A change of a communication status sets a status flag. The status flag is only reset when the status is being read. The Listener's operations will only be invoked on the communication statuses for which they are enabled by the mask. This invocation is based on the listener own status changes and/or on the status changes of the Listeners inherited from. Each bit in the bit-mask represents one of the statuses that can trigger the response of the Listener to the specified status change.

To access the Listeners all the entities define a generic operation and a specific subclass operation to access the class listener.

Example: *the DDS_DataReaderListener structure of Pointers implement as call back functions:*

```
#include "dds_dcps.h"
static struct DDS_DataReaderListener msgListener;
DDS_FooDataReader FooDR;
/* at this point, it is not important how to
create the FooDR*/
DataWriterListenerData UserDefined_ListenerData;
/* at this point, it is not important how
UserDefined_ListenerData is implemented. This
parameter can be used for Listener identification.
If not used, the parameter may be NULL. */
/* Prepare a listener for the Foo DataReader. */
msgListener = DDS_DataReaderListener__alloc();
msgListener.listener_data =
UserDefined_ListenerData;
msgListener.on_requested_deadline_missed = NULL;
msgListener.on_requested_incompatible_qos = NULL;
msgListener.on_sample_rejected = NULL;
msgListener.on_liveliness_changed = (void (*)(void
*, DDS_DataReader)) on_live_change;
msgListener.on_data_available = NULL;
msgListener.on_subscription_match = NULL;
msgListener.on_sample_lost = NULL;
/* Set the Listener with a mask only to trigger on
on_liveliness_changed. */
status = DDS_DataReader_set_listener (FooDR,
&msgListener, DDS_LIVELINESS_CHANGED_STATUS);
```

This example presents the allocation and initialization of a DDS_DataReaderListener which is only enabled for the status on_liveliness_changed. The Listener msgListener will be attached to the created DDS_DataReader named FooDR. As we can see, we have associated to the pointer msgListener.on_liveliness_changed a call back function named on_live_change that will be triggered if the status is matched.

VII. FLEXRAY DRIVER UNDER µC-OSII

For our research studies, we developed a FlexRay driver under the µC-OSII Real-Time Operating System and some

Phycore PCM023 cards. We added for each a daughter card containing the Fujitsu MB88121C component. The driver behavior, as shown on Fig. 5, is based on the communication between the ISR (Interrupt Service Routine) and *FlexRayTx/ FlexRayRx* Tasks that manage the communication process. Note that the *FlexRayTx* task has the publisher as type where the *FlexRayRx* has the subscriber as type.

- **FlexRayReceiveTask ~Subscribtion task**: the *FlexRayReceiveTask* pends (reads from the mailbox) for the received message in its mailbox posted by the corresponding Interrupt Service Routine (ISR). When a data sample FlexRay Frame arrives, this task is responsible for the deserialization (extracting Frame-ID) and for storing the data in the receive queue of the *DataReader*.
- **FlexRayPublishingTask (only one by publisher):** *Every* user task calling Write () operation may use a semaphore that will lock the task when the *DataWriter*'s send queue is full. The Frames are transmitted using one of the two following modes.

Synchronous Publishing Mode: the user task invokes the *DataWriter*'s Write() operation which puts the samples (FlexRay Frame) on a separate "queue" and then calls the Write() operation within the FlexRay API Driver to put the frame in the FlexRay controller’s transmit buffer before returning to the user task.

Asynchronous Publishing Mode: In this mode the Write () operation returns immediately to user task leaving the corresponding ISR to transfer frame from the queue to the controller transmit buffer. However, a flow controller (a separate task) is needed to reorganize the transmit queue depending of the FlexRay frame ID.

The TX/RX tasks react to the messages posted by the ISRs according to the related mailbox; in fact each mailbox is associated with an interruption event. Since FlexRay has two buses (channel) that can either send or receive the data frame, four mailboxes are needed to represent these communication events.

For example, if the *FlexRayRx* task receives an interruption on MailBox 2 (MB2) it knows that the related event is that the reception buffer is full and so calls the Read () function to read from the reception buffer.

The communication process using Mail Boxes under µC-OSII is driven by the OS_MBPend() and OS_MBPost() operations, as shown in Fig. 6, The ISR posts the messages on the mailboxes and the *FlexRayRx/Tx* task pends them.

The Write () and Read () operations prototypes are written as follow:

CR=Write (descriptor, @Buffer, size)
CR=Read (descriptor, @Buffer, size)

Where the descriptor indicates the channel that the event is associated to, the descriptor is known by task according to the Mailbox number (the OS_Event pointer) it refers to the occurred event that caused the interruption. The buffer address is the user buffer address where data should be written to and from. And finally, the size argument is an optional argument describing the user data size.

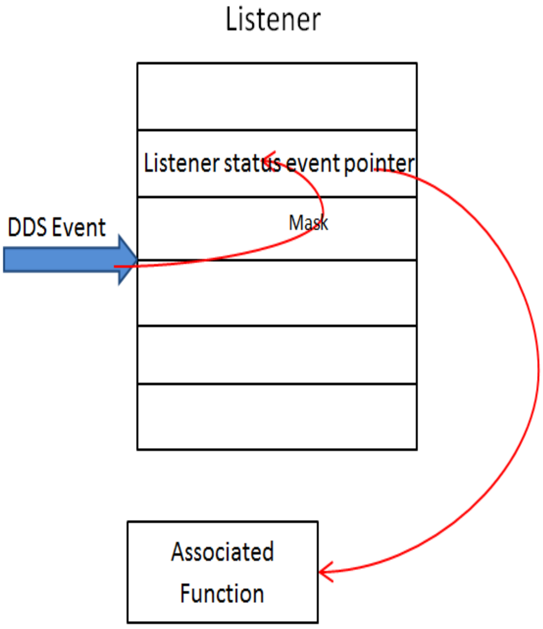


Figure 4. Listener implement as callback function

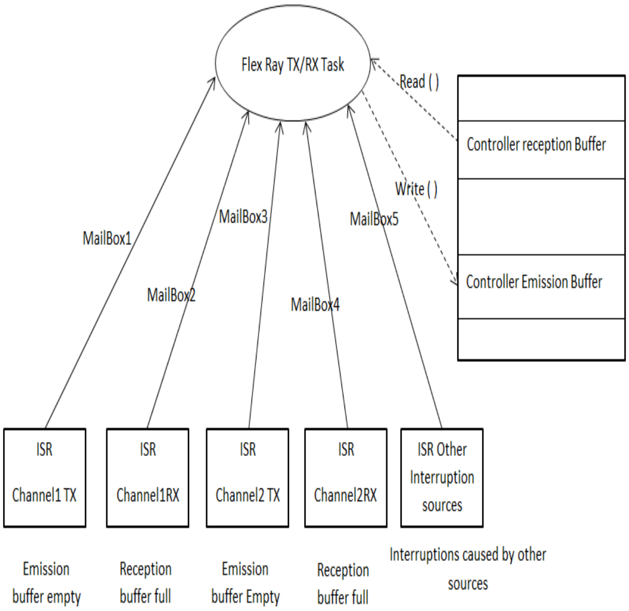


Figure 5. FlexRay Driver under µC-OSII

The returned value CR indicates whether the write/Read operation was successful or if an error occurred during the process and even the nature of the error.

If the ISR indicates the arrival of a frame, the *FlexRayRx* task will call the Read () operation, but if the ISR indicates that the emission buffer is empty the *FlexRayTx* task will call the Write () operation to write into the controller emission buffer.

According to the value of CR the *FlexRayRx* task will decide the next step to take.

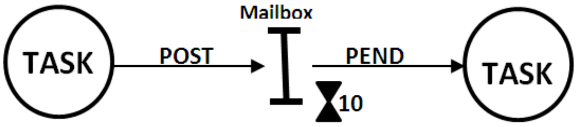


Figure 6. Message MailBox [5]

VIII. Relationship between DDS listeners and FlexRay Driver

After developing the FlexRay driver, we have noticed that for each kind of ISR the *FlexRayRx/Tx* task associates a default behavior, we will use DDS listeners to set default behavior for each ISR.
When an interruption occurs and the ISR sends the corresponding message, the *FlexRayRx/Tx* task will only get the ID from the frame. Since in FlexRay protocol an ID is usually associated to a data type we will assume that the flexRay ID is equivalent to the Topic Key in DDS. Therefore, getting the ID from the frame is the same as identifying the Topic.

The couple (ID, MB number) is now the unique identifier used by the *FlexRayRx/Tx* task to choose whether to call the Topic Publisher or the Subscriber. Actually, the MB number helps identifying if the event is a received data or an empty space in emission buffer, and the ID represents the Topic identifier.

A. *Subscription Case*

The subscription is related to the ISR event "controller's reception buffer full", in this case after identifying the event and the topic, the *FlexRayRx* task will call the appropriate *DDS_Subscriber* related to the identified Topic. In fact, it's the *DDS_SubscriberListener on_data_on_readers* operation that is called. This listener is identified by *FlexRayRx* task thanks to the pointer *listener_data*, attribute that can be used to supply the identity of the Listener.

This operation will then search for the linked list representing the *DDS_DataReaders* corresponding to the Topic. Since a *DDS_DataReaderListener* is attached to each *DDS_DataReader*, while browsing the linked list the *DDS_DataReaderListener's* operation *on_data_available* will be triggered on each listener object. The listener will have as parameter the related *DDS_DataReader* and so can call its operation *DDS_read* to get the data.

Note that if *on_data_on_readers* is called, then the middleware will not try to call *on_data_available*. However, in this case, the application will force this call and the *DDS_DataReader* objects will get data by the mean of the notification process.

Fig. 7 illustrates the whole subscription scheme representing the callback routine.

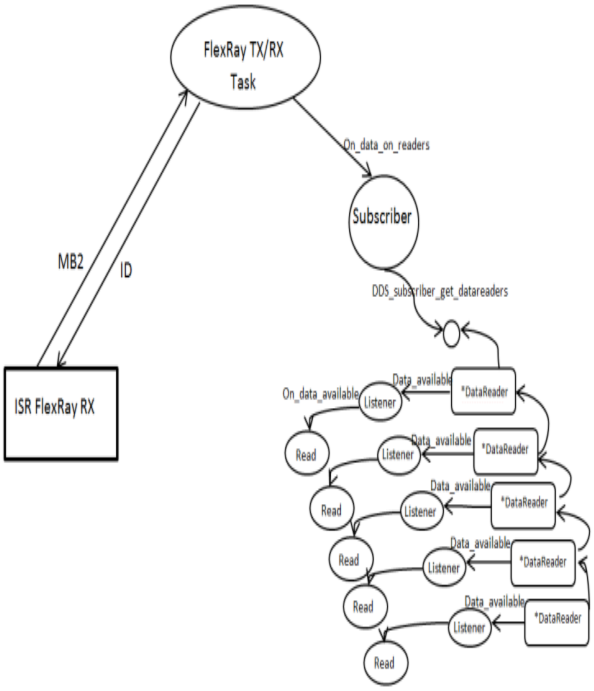


Figure 7. Subscription Routine

B. *Publication Case*

The publication is related to the ISR event "controller emission buffer empty", in this case after identifying the event and the topic, the *FlexRayTx* task will call the appropriate *DDS_Publisher* related to the identified Topic. In fact, it's the *DDS_PublisherListener on_publication_match* operation inherited from *DDS_DataWriterListener* that is called. This listener is identified by *FlexRayTx* task thanks to the pointer *listener_data*.

The Publisher will then search for the linked list representing the *DDS_DataWriters* corresponding to the Topic. Since a *DDS_DataWriterListener* is attached to each DDS_DataWriter, while browsing the linked list the DDS_DataWriterListener's operation *on_publication_match* will be triggered on each listener object. The listener will have as parameter the related *DDS_DataWriter* and so can call its operation *DDS_write* to write the data into the buffer.

Note the DDS specification does not set a default behavior in the Publication case, but since the use of Listener is a given option we have set our own default publication behavior matching the FlexRay Driver Needs.

Fig. 8 illustrates the whole publication scheme representing the callback routine.

IX. Conclusion

The Real-Time Middleware DDS offers a communication model between application level and physical layer. One of the rational uses of this middleware would be its association with a real-time network such as CAN (Control Area Network) or FlexRay Networks so that we increase the networks performances related to response time.

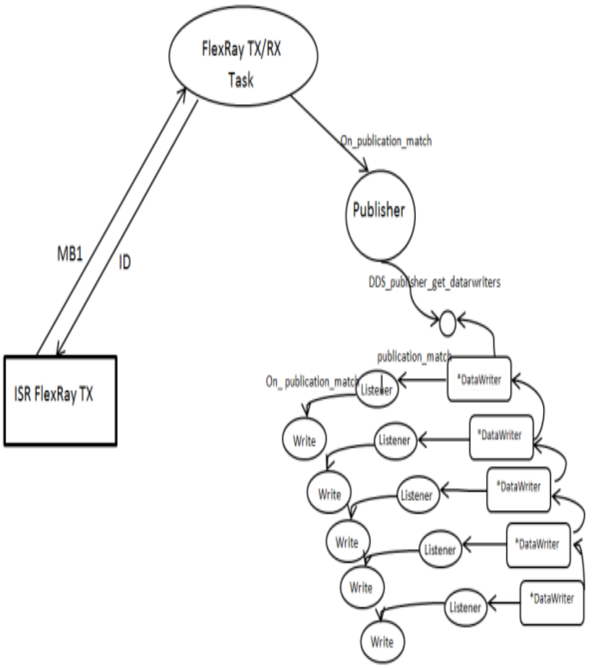


Figure 8. Publication Routine

The aim of this work is to use DDS Listeners to define a default response behavior for FlexRay ISR so that each time an interruption related to communication status occurs, a default routine is called. In this purpose we have used the DDS Listeners based on callback functions to link each ISR with the appropriate routine to replace the usual FlexRay driver's read and write operations.

The defined routine for the subscription process is indicated according to the DDS specification, but for the publication routine the DDS specification does not set a default behavior so we had to implement one of our own to match the FlexRay driver needs.

In the future works we will use this association (FlexRay Network and DDS middleware) to study its performances on a vehicle network based on the SAE (Society of Automotive Engineers) benchmark network model. This model will contain 13 nodes and applications reading and writing on the FlexRay buffers and using for the first time the DDS middleware instead of the usually used in the automobile field, the middleware AUTOSAR. We have already developed the SAE benchmark model and added the node number 13 to it, and its validation has been made using the Tasking compiler and PHYTEC XC167 cards [11].

ACKNOWLEDGMENT

The research presented in this paper would not have been possible without the support of our colleagues. We wish to express our gratitude to the SYSCOM ENIT members for their help and assistance.

REFERENCES

[1] Object Management Group- Manufacturing Domain Task, Data Acquisition from Industrial Systems specification, OMG document dtc/01-09-03, November 2002. http://www.omg.org/technology/documents/recent/omg_manufacturing.htm. 20.08.2011.

[2] Prism Tech, "Open splice C reference guide", version 2.2, Massachusetts: Burlington, 2006, pp. 22-25.

[3] N. Wang, D. Schmidt, H. Van't Hag and A. Corsaro, "Toward an adaptive data distribution service for dynamic large-scale network-centric operation and warfare (NCOW) systems", IEEE, pp. 2-3, August 2010.

[4] Object Management Group, "Data distribution service for real-time systems", version 1.2, Massachusetts: Needham, January 2007, pp. 129-130.

[5] JJ. Labrosse, "µcOS-II the real time kernel", Kansas: Lawrence, November 1998, pp. 6-7.

[6] RTI and R. Joshi, "Achitecting high performance distributed real-time applications with Java", April 2007.

[7] Prism Tech, "Open splice C++ reference guide", version 2.2, Massachusetts: Burlington, 2006.

[8] DDS vendors, http://portals.omg.org/dds/category/web-links/vendors. 20.08.2011.

[9] Open DDS, http://www.opendds.org/. 20.08.2011.

[10] T. Guesmi, R. Rekik, S. Hasnaoui, and H. Rezig, "Design and performance of DDS-based middleware for real-time control systems", IJCSNS Vol.7 No.12, December 2007, pp. 188-200.

[11] H. Jaouani, R. Bouhouch, W.Najjar, and S.Hasnaoui, "DDS on top of FlexRay vehicle network", IEEE- VCN 2011, unpublished.

Security and Performance Analysis of IPsec-based VPNs in RSMAD

Marcin Sokol, Slawomir Gajewski, Malgorzata Gajewska, Leszek Staszkiewicz
Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology
11/12 G. Narutowicza Str., PL-80-233 Gdansk, Poland
e-mail: {marcin.sokol, slawomir.gajewski, malgorzata.gajewska, leszek.staszkiewicz}@eti.pg.gda.pl

***Abstract*—The paper discusses the security architecture of the Radio System for Monitoring and Acquisition of Data from Traffic Enforcement Cameras with particular emphasis on the structure of the security of network layer of the system. The security of this layer of RSMAD is provided mainly basing on Virtual Private Networks. To implement a VPNs in RSMAD IPsec Encapsulation Security Payload in tunnel mode have been used. Data protection mechanisms and the type and parameters of the VPNs used in RSMAD have been selected on the basis of simulation results presented in the paper. Analysis of results shows also that the ESP protocol is a bit less efficient than Authentication Header protocol, which is obviously related to the fact that the ESP protocol supports data encryption. The paper also discusses some advanced solutions for communications and computation used in the RSMAD system.**

Keywords-AH; ESP; IPsec; RSMAD; VPN

I. INTRODUCTION

Radio System for Monitoring and Acquisition of Data from Traffic Enforcement Cameras (in short RSMAD) is an integrated (in terms of its functions) ICT system which uses for data transmission the radio technologies available on the market.

RSMAD is primarily used for transmission, archiving, analysis and processing of data of traffic infractions from traffic enforcement cameras (in short TEC). The purpose of the construction of this system is mainly to improve road safety by reducing the number of offences and their victims. RSMAD will significantly improve the work of the police and other departments responsible for traffic control. In this context, the system belongs to the class of the most substantive and technological advanced systems dedicated for the services dealing with TECs. In general, the performance of the RSMAD system is focused on transmission of violations recorded by the TECs to Data Acquisition Center (in short DAC). In RSMAD the data is being transmitted as cryptographically secured data blocks (data is encrypted and signed digitally). Currently the data transmission is performed via public mobile GSM systems *(Global system for Mobile Communications)*, UMTS *(Universal Mobile Telecommunications System)*, police trunked networks TETRA *(TErrestrial Trunked Radio)* as well as the Internet. In the future the use of networks based on LTE *(Long Term Evolution)* or LTE-Advanced will be possible [1]. RSMAD belongs to a group of systems with distributed structure. Equipment used in the system use dedicated software, allowing sharing system's resources. Unfortunately, in distributed systems additionally using public networks for transmission of data, the data security is a serious problem. This is because such networks are significantly exposed to the activity of intruders. Secure communication based on Virtual Private Networks (in short VPN) constitutes one of the key elements of RSMAD. For the implementation of VPNs developers of the system used mainly IPsec *(Internet Protocol Security)*, which is widely regarded to be the safest way to create VPN, which has been clearly confirmed by the authors' simulation studies.

II. RSMAD'S ARCHITECTURE

Data security in the RSMAD system will be ensured through the use of advanced security mechanisms such as: confidentiality, availability and integrity. **Already at the conceptual stage, the following, crucial for the future architecture of RSMAD assumptions has been made:**

- Security of information in RSMAD is a primarily consideration in relation to performance.
- Communication is only allowed with network devices which comply with strict security policies adopted.
- Due to the nature of data transmitted and processed in RSMAD, there is a real risk of loss of confidentiality.
- Implementing data protection mechanisms in RSMAD can not impede the work of its users.

A simplified architecture of the RSMAD system's security including VPN tunnels are shown in Fig. 1 (detailed architecture of the RSMAD system is presented in [2][3]). The concept of RSMAD is to use public telecommunications networks (in particular cellular) for data transmission. Public telecommunications networks are inherently far more vulnerable to all kinds of risks than other networks. The basic threats to the RSMAD system should include: *sniffing*, *spoofing* as well as *session hijacking*. Therefore, the use of effective and reliable data protection mechanisms in the system is particularly important.

Each of systems used for data transmission (GPRS, EDGE, UMTS, HSPA TETRA), uses different security mechanisms eliminating or reducing various risks in varying degrees. Therefore, it has been decided that the RSMAD system will be equipped with additional, independent form data transmission technology, mechanisms protecting from the threats associated with data transmission via public cellular networks.

ISBN: 978-1-61208-161-8

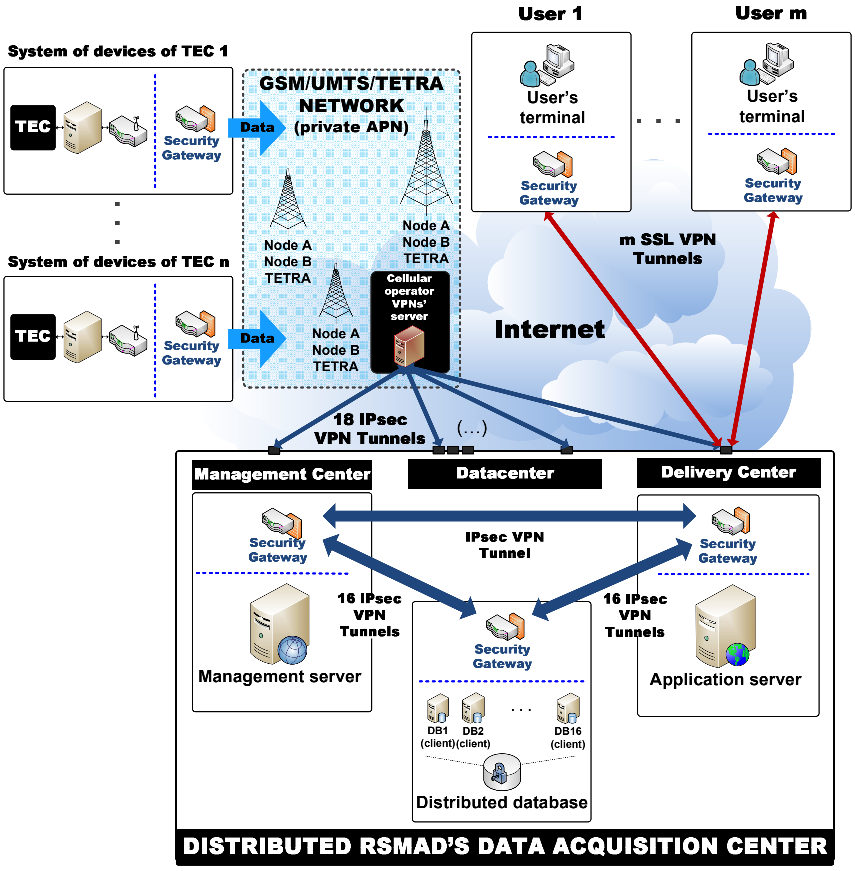


Figure 1. Simplified security architecture of RSMAD.

Thus, data protection in the RSMAD system is achieved through:

- Creating logical tunnels between the GGSN node *(Gateway GPRS Support Node)* and DAC, in a private APN subnet *(Access Point Network)* separated in the infrastructure of the GSM/UMTS operator.
- Setting data transfer limits on SIM/USIM cards *((Universal) Subscriber Identity Module)* in each location.
- Use of packets filtering and virus protection mechanisms as well as intrusion detection and prevention systems.
- Use of encryption and verification of data integrity mechanisms which are independent from the operator.

In a basic variant data will be encrypted using the *AES-128 (Advanced Encryption Standard)* algorithm and digitally signed using the *SHA-1 (Secure Hash Algorithm 1)* hash function. However, it should be noted that the security of data transmitted via public networks requires efficient performance of all securing mechanisms.

III. IP SECURITY: ARCHITECTURE AND BASIC COMPONENTS

IPsec protocol has been created through the efforts of the IPsec Protocol Working Group, being part of the IETF *(Internet Engineering Task Force)*. The primary purpose for which the group has begun work on the IPsec protocol was the supplementing the functionality of IP mechanisms to ensure the security of data transmitted using the same protocol. At the moment, the support for IPsec is one of the requirements of IPv6. In IPv4, the extension of the functionality offered by IPsec is optional. Cryptographic techniques used in IPsec, provide security to transmitted data at the level of the third layer of the ISO/OSI reference model.

IPsec protocol, by using different algorithms and cryptographic protocols provides three basic aspects of information security:

- Confidentiality.
- Integrity.
- Authentication.

There are two separate protocols in the IPsec protocols group, namely: AH *(Authentication Header)* and ESP *(Encapsulation Security Payload)*. AH protocol provides authentication using a string datagram message authentication MAC *(Message Authentication Code)*. IPsec AH protocol does not ensure the confidentiality of data (data is not encrypted). ESP provides protecting the integrity and authentication of datagrams and, in addition, their encryption. It should be noted, however, that in ESP protocol authentication and encryption services are optional.

After testing of performance of AH and ESP protocols, RSMAD has been equipped with ESP protocol, despite the fact that its efficiency was slightly lower than the AH (on average about 15%), which is shown by the results of simulation presented in Table I and Table II. Following the recommendations of [4] and [5] AH protocol provides integrity of transmitted data, by calculating and adding checksum to each datagram. In case of AH the checksum value is calculated for the entire package (including IP header). AH protocol provides also effective protection against so-called „attacks by repetition". Protection against such attacks is achieved by attaching to each datagram, the next number (the serial number of the datagram).

Recommendations [4] and [6] and its subsequent recommendations [5] and [7] describe two possible modes of operation of AH and ESP protocols: transport and tunnel. In transport mode IPsec header (AH or ESP) is placed in the IP datagram directly before the header of transport layer protocol (e.g. TCP). In turn, in tunnel mode the IP datagram (IP header with data) is firstly placed in the encrypted portion of the data, and only then followed by the addition of IPsec header (AH or ESP) and the new IP header. It should also be noted that the datagrams transmitted using the ESP have a much more complicated structure than the AH datagrams. This complexity is primarily due to the fact that the ESP protocol provides confidentiality of transmitted data by using encryption. Recommendations [6] and [7] provide for such the use of block ciphers.

Parallel use of encryption algorithms and hash function is recommended, the more that the latter characterize with only marginal use on the processor, which has been clearly confirmed by the study conducted within the RSMAD project.

It has been decided to use in RSMAD the IA *(Integrated Architecture)* implementations of IPsec protocol also called an implementation of the IPsec protocol in TCP/IP stack. It is used both in hosts and gateways. It allows to ensure the end-to-end safety. In this case, IPsec is implemented, along with the IP protocol at the level of internet layer. The use of IPsec in this implementation does not require modification of the application but interference in the IP protocol itself. The advantage of this solution is undoubtedly the fact that it supports all IPsec modes.

IV. Performance and Security Analysis of IPsec Protocol in RSMAD

A. Introduction

The aim of this study was to evaluate the performance of IPsec protocol on various configuration parameters of the channel used for secure transmission of packets via private network. For all tests, the key exchange parameters of the channel, through which data associated with the authentication and encryption (keys) are transmitted, have remained constant. Fig. 2 shows a pictorial diagram of the laboratory station.

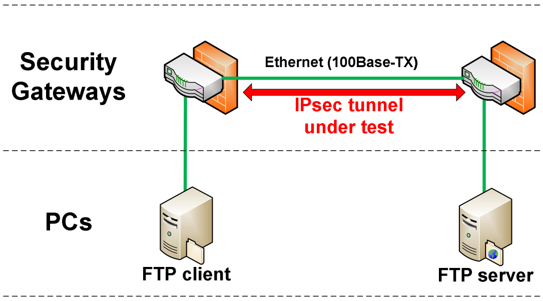


Figure 2. Simplified structure of laboratory station.

IPsec VPN tunnel was compiled between the security gateways ZyXEL ZyWALL 2 Plus. To exchange files between computers FTP *(File Transfer Protocol)* has been used. Technical characteristics of test scenarios are presented below:

1) ***Phase 1 of IPsec (permanently set):***
   - Encryption algorithm: *DES*
   - Hash function: *SHA-1*
   - Keys' exchange protocol: *Diffie-Hellmann's*
2) ***Phase 2 of IPsec (test scenarios):***
   a) ***test I parameters:***
   - Number of transmitted files: *500**
   - Total size of files: *320[MB]*
   - Device under test: *ZyWALL 2 Plus***

   ** files coming from traffic enforcement camera saved in the JPG format*
   *** max. VPN performance of ZyWALL 2 Plus is 24[Mbps]*

   b) ***test II parameters:***
   - Number of transmitted files: *1*
   - Total size of files: *320[MB]*
   - Device under test: *ZyWALL 2 Plus***

Parameters of phase 2 (data exchange) has been shown in Table I and Table II (columns 2 and 3).

B. Simulation Results

Results of simulation studies have been shown in Table I and Table II (columns 4 and 5).

The results of the studies show that the most efficient implementation of the IPsec protocol is implementation using the *DES* encryption algorithm and the *SHA-1* function for data integrity verification. As for the encryption algorithms, the least efficient algorithm is *AES-256*, although the differences in transmission are not very clear and are only about a few percent. Therefore, it seems that the selection of a set of cryptographic parameters VPN tunnels should be decided mainly for security reasons.

TABLE I. PERFORMANCE OF IPSEC PROTOCOL IN TUNNEL MODE

Type of IPsec	Type of hash algorithm	Type of cipher	Average data transfer rate in [Mbps]	
			Test I	**Test II**
ESP	SHA-1	**DES**	*18,80*	*21,44*
		3DES	*17,12*	*19,36*
		AES-128	*18,48*	*20,64*
		AES-256	*18,00*	*20,32*
	MD5	**DES**	*18,56*	*21,36*
		3DES	*16,64*	*19,60*
		AES-128	*18,08*	*20,88*
		AES-256	*17,92*	*20,24*
AH	SHA1	-	*21,28*	*22,48*
	MD5	-	*21,04*	*22,08*

TABLE II. PERFORMANCE OF IPSEC PROTOCOL IN TRANSPORT MODE

Type of IPsec	Type of hash algorithm	Type of cipher	Average data transfer rate in [Mbps]	
			Test I	**Test II**
ESP	SHA-1	**DES**	*19,24*	*21,93*
		3DES	*17,98*	*19,89*
		AES-128	*18,99*	*21,12*
		AES-256	*18,49*	*20,89*
	MD5	**DES**	*19,01*	*21,86*
		3DES	*17,03*	*20,20*
		AES-128	*18,60*	*21,34*
		AES-256	*18,49*	*20,74*
AH	SHA1	-	*21,88*	*22,99*
	MD5	-	*21,60*	*22,68*

As for encryption algorithms, the least efficient is *Triple-DES*, although differences in achieved transmission rates are not very clear and are only about a few percent. Therefore, it seems that the choice of a set of cryptographic parameters of VPN tunnels should be decided mainly by objective safety considerations which militate strongly in favor of the *AES* algorithm. Analysis of results shows also that the ESP protocol is slightly less efficient than the AH protocol, which is obviously related to the fact that the ESP protocol supports data encryption. Regularity can be noticed, that transmission of large files runs more efficiently (test I and test II scenarios). It results from restrictions of the FTP protocol used in the experiment. Research has also shown that introduction of additional (except VPN) mechanisms for data protection in a security gateway will cause additional (over 10 percent) reduction in the efficiency of the VPN network.

V. CONCLUSION AND FUTURE WORKS

IPsec can be undoubtedly considered as a very solid mechanism that allows the removal the imperfections and drawbacks of the IP protocol in the aspect of security. IPsec is currently agreeably recognized by most experts as the best mechanism to implement VPN in terms of security.

Without any doubt, the fact that the IPsec implementations exist for virtually all operating systems also speaks in favor of IPsec protocol. Definitely the most popular of these is FreeSWAN, designed for family of Linux operating systems.

In conclusion, we can acknowledge that IPsec is now the safest way to create a VPN network. Decreasing interest in this protocol, observed recently, is primarily due to the fact that SSL VPNs are much simpler to implement. Probably the role of SSL in telecommunications will consistently grow. However, despite a few flaws IPsec makes the impression of the best proposal, being far ahead of competitive solutions in terms of scalability and security.

Studies conducted in the RSMAD project have shown that the IPsec protocol is probably the best security protocol currently available. Similar analysis concerning other designed for similar purposes have shown that none of them was perfect. On one hand, IPsec is much better than any of the IP security protocols developed in recent years. On the other hand it seems to us that it will never lead to the creation of a fully secure system. The use of *AES* algorithm in IPsec protocol brings very tangible benefits: on the one hand it increases the security of transmission, on the other hand, the network provides a satisfactory efficiency. Therefore, it is worth noting that manufacturers of devices using IPsec protocol and *AES* need not incur any licensing costs. This affects very positively the dissemination of the *AES* algorithm, because it does not lead to an increase in prices of such devices and applications.

Until the appearing the IPsec implementations supporting *SHA-2* algorithms, we should we decide to use the implementations supporting *SHA-1* functions. Using the *MD5* function could realistically threaten the integrity of data transmitted via IPsec. In favor of *MD5* speaks only a significantly higher performance compared to the function of the SHA family (what is interesting, including *SHA-2*).

In view of the results of studies and security requirements for RSMAD, it was decided to use the IPsec ESP version (*AES-128*, *SHA-1*), in tunnel mode and implementation of the IA.

ACKNOWLEDGMENT

This research work is carried out under research and development grant No. N R02 0034 06 in 2009-2012, in the Department of Radiocommunication Systems and Networks, Faculty of Electronics, Telecommunications and Informatics in Gdansk University of Technology. The work is financed by the National Centre for Research and Development.

REFERENCES

[1] KSSR DT 07.100 v. 1.0.1, General concept of RSMAD's DAC (in Polish), Gdansk University of Technology, Poland 2009.

[2] KSSR RT 02.902 v. 1.1.0, Technical architecture for cryptographic security of RSMAD (in Polish), Gdansk University of Technology, Poland 2009.

[3] KSSR RT 02.901 v. 1.1.0, Security architecture of RSMAD system (in Polish), Gdansk University of Technology, Poland 2009.

[4] R. Atkinson, RFC 1826: IP Authentication Header, 1995.

[5] S. Kent and R. Atkinson, RFC 2402: IP Authentication Header, 1998.

[6] R. Atkinson, RFC 1827: IP Encapsulating Security Payload (ESP), 1995.

[7] S. Kent and R. Atkinson, RFC 2406: IP Encapsulating Security Payload (ESP), 1998.

Using DRBL to Deploy MPICH2 and CUDA on Green Computing

Jiun-Yu Wu[12], Yao-Tsung Wang[2], Steven Shiau[2], Hui Ching Wang[1]
[1]Department of Applied Mathematics, National Chung Hsing University, Taichung City, Taiwan.
[2]National Center for High Performance Computing, Taiwan.
e-mail: adherelinux@hotmail.com, jazz@nchc.org.tw, steven@nchc.org.tw, hcwang@amath.nchu.edu.tw

***Abstract*—In this paper, an energy efficient architecture for Build Energy Efficient GPU and CPU Cluster Using DRBL is proposed. This architecture helps administrator not only to quickly deploy and manage GPU and CPU Cluster environment, but also bring benefit of energy efficiency in scientific computing. The experiment simulates 3 cases to prove energy efficiency. We will compare GPU and CPU Cluster Using DRBL with the design without DRBL. According to the experiment results, the architecture provides a way to implement a power economization computing architecture that reduces energy consumption.**

Keywords-Green Computing; CUDA; DRBL; MPICH2; GPU.

I. INTRODUCTION

HPC (High Performance Computing) is an important research domain for all of human; it helps people to solve science questions massively. On the other side, it also brings energy consumption and environmental problems. In recent years, both green computing and grid computing are putting together for discussion, and energy consumption optimization becomes a crucial issue to grid computing instead of computing performance.

Education institutions or research organizations that demand to adopt Cloud Computing to solve computing and energy saving problems may use our toolkit for practice or experiment. DRBL helps a lot on energy saving and cost down because of the diskless design. The DRBL provides a diskless or systemless environment for client machines. It works on Debian, Ubuntu, Mandriva, Red Hat, Fedora, CentOS and SuSE. DRBL uses distributed hardware resources and makes it possible for clients to fully access local hardware. We will demonstrate how DRBL really works on power saving and how much energy DRBL can save.

Cloud computing is more and more popular recently owing to its characteristics such as distribution file system [2, 3], implementation of large-scale tasks, analyzing very large data sets, etc. As we implement large-scale tasks, more power is consumed. Green computing is a growing topic in recent years. Our objective is to build an energy-saving environment by DRBL software.

Green computing is environmentally responsible use of computers and related resources [6, 7]. The goal of green computing is efficient resource utilization as well as reduced resource consumption [8]. Some suggested practices for Green Computing are the following ways: (1) find out how much energy in IT system; (2) ensure unused equipment are turned off when it is not being used; (3) educate staff to the benefits of saving energy and recycling; and (4) identify IT management practices that reduce power consumption [6]. Some approaches of green computing are algorithmic efficiency; virtualization, terminal server, and power management (including power supply, storage, and other devices).

The architecture is composed of DRBL (Diskless Remote Boot in Linux) [1] MPICH2 (High-performance and Widely Portable MPI) [5], and CUDA (Compute Unified Device Architecture) [15] software. DRBL server offered clients to use MPICH2 and CUDA software. Clients don`t install any software. Clients use PXE to connect the server and manage the deployment operating system. We are necessary installation CUDA driver and MPICH2 library on the server. Figure 1 shows system environment, and clients can run MPI program and CUDA program.

II. BACKGROUND

A. DRBL

Diskless Remote Boot in Linux (DRBL) is an open source solution to managing the deployment of the GNU/Linux operating system across many clients [9]. DRBL supports lots of popular GNU/Linux distributions, and it is developed based on diskless and systemless environment for client machines. Figure 1 shows DRBL system architecture. DRBL uses PXE/Etherboot, DHCP, TFTP, NFS and NIS to provide services to client machines, so it is not necessary to install GNU/Linux on the client hard drives individually. Users just prepare a server machine for DRBL to be installed as a DRBL server, and follow the DRBL installation wizard to configure and dispose the environment for client machines step by step. It's really an easy job to deploy a DRBL environment on clustering systems even for a GNU/Linux beginner. Consequently, cross-platform and user-friendly are the key factors that make the DRBL become a superior clustering tool. DRBL can efficiently deploy diskless or diskfull cluster environment, and manage client. It configures these services (TFTP, NIS, DHCP, and NFS) to build a cluster environment. According to this implementation, an administrator just needs two steps to deploy cluster environment. (1) Step 1:

Installs DRBL packages and generates kernel and initrd for client; (2) Step 2: setup environment parameters, such IP address, and numbers of clients. It also provides cluster management and cluster system transformation (diskfull or diskless system).

B. CUDA

In the recent years, the graphic card has become powerful computing tools. The CUDA software created by NVIDIA Company, CUDA was parallel computing architecture Graphics processing units (GPUs) were originally designed to perform the highly parallel computations required for graphics rendering. The massive parallel computing ability has made GPUs very powerful devices in solving scientific and engineering problems recently. GPU is programmable level upgrade due to the great improvement of Semiconductor Processing technology.

General single GPU (Graphics Processing Unit) usually contains hundreds of programmable processing units, e.g., NVIDIA GEFORCE 9800 GT has 112 processors. It has powerful computation capability for engineering scientific problems. GPUs can handle for acceleration of image processing, linear algebra computations, image processing, molecular dynamics, ray tracing and so on.

In 2006, NVIDIA introduced a general-purpose parallel computing architecture, CUDA (Compute Unified Device Architecture). CUDA is a new parallel programming model that supports C programming language, FORTRAN, and now, OpenCL. C++ will soon be supported in the future according to the announcement of NVIDIA. CUDA gives instruction set for the parallel computation in CUDA GPUs. CUDA can solve data-intensive computing.

C. MPICH2

MPICH2 [4] is a tool of the Message-Passing Interface for CPU. MPICH2 was proposed by Argonne National Lab. MPICH2 is open source which is freely available license.It support system, including Microsoft Windows, UNIX and Linux (Ubuntu, Centos, Fedora, etc.). The latest of version is 2-1.0 that we can download on official website.MPICH2 is implementation for distributed-memory and shared memory in parallel computing. MPICH2 offer parallel programming library which supports C, C++, Fortran language. The MPICH2 offers us some library which uses very convenience. In this paper, we use DRBL deploy to MPICH2 and CUDA software on PC cluster.

III. ARCHITECTURE OVERVIEW FOR GPU AND CPU CLUSTER

A. GPU and CPU Cluster

We have 8 computers in the experiment environment. We combine DRBL with MPICH2, CUDA in my system environment. People can use GPU and CPU clustering quickly because DRBL helps to easy and quickly deploy GPU and CPU clustering environment. The project is suitable for PC classroom. The reasons are as the following. (1) PC Classrooms need the same system and centre management; (2) It allows all the configuration of your client computers by installing just one server machine; (3) Only the server has hard disk, ,and client has not hard disk; (4) Compared with diskfull design, the diskless design saves power consumption in PC classrooms.

Figure 1 shows system architecture. At least 20 clients in PC classroom can be easily managed by the DRBL server. Each client has 4 cores in CPU cluster. Each client gets a NVIDIA GEFORE 9800GT graphics card which supports CUDA. Clients are diskless machines. Users just prepare a server machine for DRBL installation as a DRBL server. They will boot via PXE/Etherboot. Users will find that all of these CPUs and memory of nodes will merge by DRBL server. We configure installation software in the DRBL server before we arrange clients. Users just prepare a server machine for DRBL installation wizard to configure a push the environment for client machines step by step. DRBL can support user to massively deploy and effectively manage clusters.

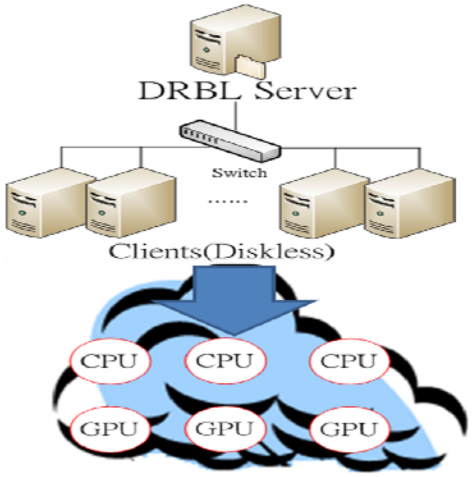


Figure 1. Using DRBL on GPU and CPU Cluster

It is efficient to deploy diskless cluster environment by DRBL component. In the first step, it has to install DRBL software in the server. If the OS is Debian or RPM package system, only 1 package from DRBL website should be installed. Then it needs to execute DRBL configuration command “drplsrv –i” to choose your Kernel version for nodes and automatically installs the packages that DRBL required, such as DHCP, NFS, NIS and TFTP. Then, using DRBL deployment command “drblpush –i” to push system environment to all clients. DRBL offers interactive dialog to help users to build DRBL environment and it automatically configures and starts all the services required to make the Cluster work. It automatically detects the network interfaces that have private IP addresses assigned to them and asks administrator how many clients will be setup. DRBL provides two methods for nodes IP address: (1) fixed IP address (binding MAC address): this feature is useful to setting up system for security; (2) dynamic IP address (range of IP address) in the open environment where anyone can add a new machine Experiment Cases.

B. *Experiment Environment*

The experiment uses computers in PC classroom in our research center. One of computers has already DRBL server with software (such as: MPICH2, g++, gcc, NVIDIA-Linux-driver, cudatoolkit, cudasdk and) in the server. It's very flexible to transform between two different modes cluster environment (diskfull and diskless) through DRBL.

The cluster has 1 server, 7 clients, and the PC are equipped with Intel® Core(TM)2 Quad CPU Q9550 @ 2.83GHz. Table I illustrates Hardware specifications and Software list.

TABLE I. HARDWARE SPECIFICATIONS AND SOFTWARE LIST

Hardware (PC)	Software
Intel$^{(a)}$ Core™ 2 Quad CPU Q6600 2.4Hz	Ubuntu 10.04
8 GB RAM	Kernel 2.6.32.21
160GB Hard disk	DRBL 1.94-27
Intel 82571EB Gigabit NIC	gcc, g++, fort77, MPICH2
Hardware(Network switch)	
Linksys SLM2048 48 port 10/100 Gigabit Switch	

C. *Measured Environment*

Figure 2 shows our experimental and measured environment. The measured environment includes one DRBL server, seven clients, one network switch, one electricity monitor and one notebook for data collection. The server was installed DRBL and other software. It can transform between diskfull and diskless cluster environment easily. These one power distribution units (PDUs) supply power to these machines and transport power utilization (ampere) to server.

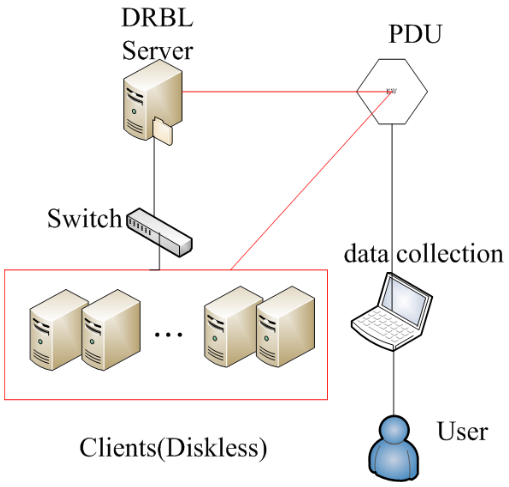


Figure 2. Energy Measured Environment Cluster

The electricity measurement has two categories on my environment. These categories are performed with two evaluation cases: (1) Examples of CUDA (2) Examples of MPICH2. CUDA uses GPU Computing SDK code samples. The MPICH2 solves Laplace equation, Gauss elimination, and performs Matrix multiplication.

IV. POWER CONSUMPTION CALCULATION AND EXPERIMENT DESIGN

A. *Amazing PDU Utility*

Figure 3 is the power evaluation (Amazing PDU Utility [10]). The major functions of Amazing PDU Utility are as the following: (1) an Ethernet interface for the built-in web server; (2) Audible Alarm (warning and overload); (3) Graphical User Interface; (4) Meter; (5) Calculation of power consumption ; (6) Graphics report output. The PDU has a total current and Kilowatt record every minute. User can observe the current variable and download relative data from electricity monitor.

Figure 3. Energy Measured Environment Cluster

B. Experiment Design

We provide three examples to test. We use GPU Computing SDK code samples. It can be download from the NVIDIA website. We do compile the GPU Computing SDK code samples in ubuntu system. The examples are briefly introduced as follows:

- An NBODY use the gravitation potential to determining inter-atomic forces. The formulation of force present by the following:

$$f_{ij} = G\frac{m_i m_j}{\| r_{ij} \|^2} * \frac{r_{ij}}{\| r_{ij} \|}$$

- PARTICLE systems are used in many simulations from Molecular dynamic and astrophysics simulations and computational fluid dynamics, etc.
- The POSTPROCESSGL example use CUDA to interoperability to post-process an image of a 3D scene generated in OpenGL.

We use these examples to test in MPICH2 cluster. The following examples have paralleled programming. The example is introduced by the following.

- Using finite difference method to solve Laplace equation. The Laplace equation is given by:

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \qquad (1)$$

$$IC : \phi(0,0) = 100$$

$$Bc : \phi(x,0) = \phi(0,y) = \phi(1,y) = \phi(x,1) = x^2 - y^2$$

By numerical discretization, substituting Eqn in the Laplace equation, we defined

$$\phi_{i,j} = \frac{1}{4}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1}\phi_{i,j-1}) \qquad (2)$$

This example handles 4096*4096 meshes on 32 processors.

- The Gauss Elimination method solves matrix equations. The algebraic equation is AX=B. This example handles 3200*3200 meshes on 32 processors.

- The MATRIX MULTIPLICATION is important for linear algebra. This example handles 4096*4096 meshes on 32 processors.

C. Power Consumption Result

Table II is Kw/h for CUDA, MPICH2 in power consumption. Table Ш shows the power saving percentage in power evaluation experiment. Table IV is execution time. NBODY example saves almost 9 % power compared to boot with hard disk, PARTICLE example saves almost 11 % power compared to boot with hard disk, POSTPROCESSGL example saves almost 5.8 % power compared to boot with hard disk, LAPLACE EQUATION example saves 3.8 % power compared to boot with hard disk, GAUSS ELIMINATION example saves 3.4 % power compared to boot with hard disk, MATRIX MULTIPLICATION example saves 4.1 % power compared to boot with hard disk.

TABLE II. ALL EXAMPLES OF POWER CONSUMPTION

CUDA(1-3)
MPICH2(4-6)
	PXE boot with DRBL server (Kw/h)	Local boot with hard disk (Kw/h)

1.NBODY	0.02889	0.03178
2.PARTICLES	0.02864	0.03225
3.POSTPROCESSGL	0.02586	0.02746
4.LAPLACE EQUATION	0.04069	0.04234
5.GAUSS ELIMINATION	0.02439	0.02525
6.MATRIX MULTIPLICATION	0.03982	0.04153

TABLE III. POWER SAVING PERCENTAGE IN POWER EVALUATION EXPERIMENT

CUDA(1-3)
MPICH2(4-6)
	Boot with DRBL design VS boot with hard disk (%)

1.NBODY	9.09377
2.PARTICLES	11.1938
3.POSTPROCESSGL	5.826657
4.LAPLACE EQUATION	3.897024
5.GAUSS ELIMINATION	3.405941
6.MATRIX MULTIPLICATION	4.117505

*Boot with DRBL design VS boot with hard disk = (Local boot with hard disk - PXE boot with DRBL server) / Local boot with hard disk * 100%.

TABLE IV. TIME CONSUMING IN POWER EVALUATION EXPERIMENT

CUDA job	PXE boot with DRBL server (s)	Local boot with hard disk (s)
NBODY	60	60
PARTICLES	60	60
POSTPROCESSGL	60	60

TABLE V. TIME CONSUMING IN POWER EVALUATION EXPERIMENT

MPICH2 job	PXE boot with DRBL server (s)	Local boot with hard disk (s)
Laplace equation	75	77
Gauss elimination	42	44
Matrix multiplication	66	70

V. DISCUSSION AND CONCLUSION

We can say that the diskless design of DRBL can bring effect on power saving for CPU and GPU applications. Table III presents all cases. It shows that the power consumption percentage in our experiment. Owing to the bottleneck of high throughput I/O data communication and overhead, the DRBL architecture is not suitable for I/O intensive applications. On the other hand, it brings a great benefit to those CPU intensive applications with RAM disk [11].

In the future, we will research into the relation between power consumption and various CPU, GPU frequencies on Ubuntu. We can use cpufreqd instruction that adjusts CPU frequency under diskfull and diskless environment. CPU frequency is dynamically controlled by cpufreqd. We can use NVClOCK software to overclock NVIDIA based video cards on the Linux system. GPU frequency is dynamically controlled by NVClOCK.

We can observe the effects of various CPU, GPU frequencies on power consumption under diskfull and diskless environment. The green computing is more and more important in the future, we still have lots of performance tuning and power measuring work for advanced evaluation. Some of the tools are used for advanced evaluation, such as bootchart [12], lm-sensors [13] and powertop [14].

REFERENCE

[1] Diskless Remote Boot in Linux (DRBL), NCHC. [Online]. http://drbl.sourceforge.net/ [accessed; Jan, 2011].

[2] J. Cope, M. Oberg, H. M. Tufo, and M. Woitaszek, “Shared Parallel Filesystems in Heterogeneous Linux Multi-Cluster Environments” in Proc. 6th LCI International Conference on Linux Clusters: The HPC Revolution, 2005.

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, 2008. GCE '08, Vol. 12-16, pp. 1-10, 2008.

[4] DeinoMPI. [Online]. http://mpi.deino.net /[accessed; Feb, 2011].

[5] MPICH2: High-performance and Widely Portable MPI [Online]. http://www.mcs.anl.gov/research/projects/mpich2/ [accessed; Jan, 2011].

[6] Green Computing, Wikipedia. [Online]. http://www.greenlivingpedia.org/Green_computing [accessed; Jan, 2011].

[7] Green Computing definition webpage on SearchDataCenter.com. [Online]. Available: http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci1246959,00.html [accessed; Jan, 2011].

[8] Green Computing webpage on Computing. [Online]. http://www.computing.co.uk/ctg/special/1814241/green-computing [accessed; Feb, 2011].

[9] W. C. Kuo, C. Y. Tu, and Y. T. Wang, "Deploy Kerrighed SSI Massively Using DRBL," HPC ASIA 2009, 2009.

[10] Amazing PDU Utility [Online]. ftp://ftp.opengear.com/ip-pdu/cd/User_Manual/PDU_Utility_User_Manual_v2.pdf [accessed; Jan, 2011].

[11] Che. Y. Tu, W. C. Kuo, Y. T. Wang, and S. Shiau, "Building Energy Efficient ClassCloud using DRBL", 10th IEEE/ACM International Conference Grid Computing, Vol. 13-15, pp. 189-195, 2009.

[12] The Bootchart website. [Online]. Available: http://www.bootchart.org/ [accessed; Nov, 2010].

[13] The lm-sensors website. [Online]. Available: http://lm-sensors.or [accessed; Nov, 2010]

[14] The PowerTop website. [Online]. http://www.lesswatts.org/ [accessed; Nov, 2010].

[15] P. Harish and P. J. Narayanan, “Accelerating Large Graph Algorithms on the GPU Using CUDA”, In High Performance Computing – HiPC 2007, Vol. 4873, pp. 197-208, 2007.

Inventory-Based Empty Container Repositioning in a Multi-Port System

Loo Hay Lee, Ek Peng Chew, Yi Luo
Department of Industrial & Systems Engineering
National University of Singapore
Singapore 119260
E-mails: {iseleelh, isecep, iseluoy}@nus.edu.sg

***Abstract*— The purpose of this study is to develop an inventory-based control policy to reposition empty container in a multi-port system with uncertain customer demand. A single-level policy with repositioning rule in terms of minimizing the repositioning cost is proposed to manage the empty container with periodical review. The objective is to optimize the parameters of the policy to minimize the expected total cost per period incurred by repositioning empty container, holding unused empty container and leasing empty container. The problem is solved by applying non-linear programming and a gradient search approach with Infinitesimal Perturbing Analysis (IPA) estimator. Numerical examples are given to demonstrate the effectiveness of the proposed policy.**

Keywords- empty container repositioning; inventory control; simulation; Infinitesimal Perturbation Analysis (IPA).

I. INTRODUCTION

In the last few decades, the containerization of cargo transportation has been the fastest growing sector of the maritime industries. The growth of containerized shipping has presented challenges inevitably, in particular to the management of empty containers arising from the highly imbalanced trade between countries. It is reported that empty container movements constitute approximately 20% of the world ports handing activity ever since 1998 [1]. Song [2] reports that the cost of repositioning empty container is just under $15 billions, which is 27% of the total world fleet running cost based on the data for 2002. If the cost of repositioning empty container can be reduced, the shipping company could increase profit and improve competitiveness. Therefore, how to effectively and efficiently manage ECs is a very important issue for shipping company and it is known as *empty container repositioning* (ECR) problem.

Much attention about ECR problem has been focused on utilizing mathematic models to solve this issue [3-7]. Mathematic models can often capture the nature of the problem, while give rise to concerns, such as requirement of a pre-specified planning horizon, sensitivity of the decisions to data accuracy and variability and implementation of the decisions in the stochastic systems [8, 9]. Recently, several authors turn to explore the inventory-based mechanism in addressing the ECR problem in the stochastic systems [2, 10, 11]. These studies demonstrate that the optimal repositioning policies are of the threshold control type, characterized by some parameters and rules, in some situations such as one-port and two-port systems. Researchers extend the above works to more general systems and focus on the implementation of threshold-type control policies [9, 12~14].

In this paper, we consider the ECR problem in a multi-port system which comprises a set of ports connected to each other and a fleet of owned containers are used to meet the stochastic customer demands. A single-level threshold policy with repositioning rule in terms of minimizing the repositioning cost is proposed to manage the EC with periodical review. The objective of the paper is to minimize the expected total cost per period, including transportation cost and holding and leasing cost by optimizing the parameters of the single-level policy. The paper is organized as follows. Section II presents the formulation for our problem. Section III describes the Infinitesimal Perturbation Analysis (IPA)-based gradient technique to solve the problem. Section IV illustrates the numerical studies. Conclusions are provided in the last section.

II. PROBLEM FORMULATION

We consider a multi-port system, consisting of ports connected with each other. A fleet of owned ECs meets exogenous customer demands, which are defined as the requirements for transforming ECs to laden containers and then transporting these laden containers from original ports to destination ports. A single-level threshold policy with periodical review is employed to manage the ECs. At the beginning of a period, the ECR decisions are made for each port, involving whether to reposition ECs, to/from which ports, and in what quantity. Then, when the customer demands occur in the period, we can use those containers that are currently stored at the port and those ECs that are repositioned to the port in the period to satisfy. If it is not enough, we need to lease additional ECs immediately from vendors. We make the following assumptions:

- The owned container fleet is fixed.
- Short-term leasing is considered and the quantity of the leased ECs is always available in the port at any time.
- The leased ECs are not distinguished from owned container.
- Twenty-foot equivalent unit (TEU) is used to represent a container.
- The travel time for each O-D pair (p, m) (from port p to port m) is less than one period length.
- When the repositioned ECs arrive at the destination ports, they will become available immediately.

- When the laden containers arrive at the destination ports, they will become empty and be available at the beginning of next period.

A. *Notation*

To formulate the problem, following notations are introduced firstly.

N	The fleet of owned empty containers
P	The set of ports
t	The discrete time decision period
P_t^S	The surplus port subset in period t
P_t^D	The deficit port subset in period t
P_t^B	The balanced port subset in period t
$x_{p,t}$	The beginning on-hand inventory of port p in period t
$y_{p,t}$	The inventory position of port p in period t after making the ECR decisions
$z_{p,m,t}$	The amount of ECs repositioned from surplus port p to the port m in period t
$\varepsilon_{p,m,t}$	The random customer demand for the O-D pair (p,m) in period t
$\mathbf{x}_t$	The vector of the beginning on-hand inventory in period t
$\mathbf{y}_t$	The vector of the inventory position in period t
$\mathbf{Z}_t$	The array of repositioned quantities for all ports
$a_{p,t}^S$	The amount of estimated EC supply for surplus port p in period t
$a_{p,t}^D$	The amount of estimated EC demand for deficit port p in period t
ω_t	The stochastic customer demands in period t, which is the array of a realization of the customer demands
$C_{p,m}^R$	The cost of repositioning an EC from port p to port m
C_p^H	The cost of holding an EC at port p per period
C_p^L	The cost of leasing an EC at port p per period
γ_p	The threshold of port p
$\boldsymbol{\gamma}$	Vector of the thresholds

To simplify the narrative, the following notations are introduced.

$u_{p,t}^O = \sum_{m\in P(p)} z_{p,m,t}$	The sum of ECs repositioned out from port p in period t
$u_{p,t}^I = \sum_{l\in P(p)} z_{l,p,t}$	The sum of ECs repositioned into port p in period t
$\eta_{p,t}^O = \sum_{m\in P(p)} \varepsilon_{p,m,t}$	The sum of exported laden containers of port p in period t
$\eta_{p,t}^I = \sum_{l\in P(p)} \varepsilon_{l,p,t}$	The sum of imported laden containers of port p in period t
$F_p(.)$	the cumulative distribution function for $\eta_{p,t}^O$
$\varphi_{p,t} = \eta_{p,t}^I - \eta_{p,t}^O$	The amount of the difference between the laden container inbound and outbound of port p in period t

It should be pointed out that $\mathbf{x}_1$ is a given state variable, i.e., the given initial on-handing inventory; while $\mathbf{x}_t$ is a decision variable for $t > 1$. The ECR decisions are made at the beginning of period t firstly. Then, the inventory position can be obtained by

$$y_{p,t} = x_{p,t} - u_{p,t}^O + u_{p,t}^I \quad \forall p \in P \tag{1}$$

After customer demands are realized and the laden containers become available, the beginning on-hand inventory for the next period can be updated by

$$x_{p,t+1} = y_{p,t} + \varphi_{p,t} \forall p \in P \tag{2}$$

Next, we present the single-level threshold policy to determine the repositioned quantities $\mathbf{Z}_t$ in period t.

B. *A Single-Level Threshold Policy*

To make the ECR decisions, a single-level threshold policy is developed, which tries to maintain the inventory position at a target threshold value $\boldsymbol{\gamma}$. More specifically, port p has a target threshold, namely γ_p; in each period, such as in period t, if the beginning on-handing inventory of port p, namely $x_{p,t}$ is greater than its threshold value, i.e., γ_p, then it is a surplus port and the quantity excess of γ_p can be repositioned out to other ports that may need it to try to bring the inventory position down to γ_p; if $x_{p,t}$ is less than γ_p, then it is a deficit port and ECs should be repositioned into this port from surplus ports to try to bring the inventory position up to γ_p; if $x_{p,t}$ is equal to γ_p, then it is a balanced port and nothing is done.

Without loss of generality, we consider the ECR decisions in period t. According to the threshold policy, three subsets, i.e., surplus port subset, deficit port subset and balanced port subset can be obtained as follows:
$P_t^S = \{i: x_{i,t} > \gamma_i\}$;$P_t^D = \{j: x_{j,t} < \gamma_j\}$; $P_t^B = \{b: x_{b,t} = \gamma_b\}$.
When either the surplus port subset or the deficit port subset is empty, we do nothing. That is, no ECs are repositioned and we can have $\mathbf{Z}_t = 0$. However, when P_t^S and P_t^D are nonempty, we can compute the amounts of EC supplies of surplus ports and EC demands of deficit ports by:

$$a_{i,t}^S = x_{i,t} - \gamma_i \quad \forall i \in P_t^S \tag{3}$$

$$a_{j,t}^D = \gamma_j - x_{j,t} \quad \forall j \in P_t^D \tag{4}$$

Then, the problem is about moving ECs from surplus ports to deficit ports in the right quantity at the least movement cost. A transportation model is formulated to solve this problem as follows:

$$\min \sum\nolimits_{i\in P_t^S} \sum\nolimits_{j\in P_t^D} C_{i,j}^R z_{i,j,t} \tag{5}$$

$$\text{s.t} \qquad \sum\nolimits_{j\in P_t^D} z_{i,j,t} \le a_{i,t}^S \quad \forall i \in P_t^S \tag{6}$$

$$\sum\nolimits_{i\in P_t^S} z_{i,j,t} \le a_{j,t}^D \quad \forall j \in P_t^D \tag{7}$$

$$\sum\nolimits_{i\in P_t^S} \sum\nolimits_{j\in P_t^D} z_{i,j,t} = \min(\sum\nolimits_{i\in P_t^S} a_{i,t}^S, \sum\nolimits_{j\in P_t^D} a_{j,t}^D) \tag{8}$$

$$z_{i,j,t} \ge 0 \quad \forall i \in P_t^S, j \in P_t^D \tag{9}$$

Constraints (6) and (7) are resource constraints. Constraint (8) implies that the amount of total exported ECs from the surplus ports is capacitated by the amount of total demands of all deficit ports; thus, we can try to bring the inventory position of each port back to its threshold level. Constraints (9) are the non-negative repositioned EC quantity constraints.

Solving the transportation model, we can obtain the repositioned quantities from the surplus ports to the deficit ports. To further complete the value of $\mathbf{Z}_t$, which involves the repositioned quantities for all ports, we set $z_{l,b,t} = z_{b,m,t} = 0 \ \forall l \in P(b), m \in P(b)$ for a balanced port $b \in P_t^B$, $z_{l,j,t} = 0 \ \forall l \in P(j)$ for a surplus port $j \in P_t^S$ and $z_{i,m,t} = 0 \ \forall m \in P(i)$ for a deficit port $i \in P_t^D$, which reflect the facts that a balanced port does not reposition in or out ECs, a surplus port does not reposition in ECs and a deficit port does not reposition out ECs, respectively.

C. The Optimization Problem

Let $J(N, \boldsymbol{\gamma})$ be the expected total cost per period with the fleet size N and policy parameter $\boldsymbol{\gamma}$. The problem, which is to find the optimal parameters of the given policy, namely $\boldsymbol{\gamma}^*$ that minimizes the expected total cost per period can be formulated as

$$\min_{\boldsymbol{\gamma}} J(N, \boldsymbol{\gamma}) \tag{10}$$

subject to the single-level threshold policy, the given fleet size N and the inventory dynamics equations (1) and (2). With a slight misuse of the notation, we drop the subscript t in the notations of $\mathbf{x}_{\mathrm{t}}, \mathbf{y}_{\mathrm{t}}, \omega_{\mathrm{t}}, \mathbf{Z}_{\mathrm{t}}$ and $\eta_{p,t}^O$ for ease of description. More specifically, $J(N, \boldsymbol{\gamma})$ can be formulated as:

$$J\left(N, \boldsymbol{\gamma}\right) = EJ\left(\mathbf{x}, \boldsymbol{\gamma}, \omega\right) = E\left(H\left(\mathbf{x}, \boldsymbol{\gamma}\right) + G\left(\mathbf{y}, \omega\right)\right) \tag{11}$$

where $J(\mathbf{x}, \boldsymbol{\gamma}, \omega)$ is the total cost in one period; $H(\mathbf{x}, \boldsymbol{\gamma})$ and $G(\mathbf{y}, \omega)$ are the EC repositioning cost and the total EC holding and leasing cost in one period, respectively. We have

$$H(\mathbf{x}, \boldsymbol{\gamma}) = H(\mathbf{Z}) = \sum\nolimits_{p \in P} \sum\nolimits_{m \in P(p)} C_{p,m}^R z_{p,m} \tag{12}$$

$$\begin{aligned} G(\mathbf{y}, \omega) &= \sum\nolimits_{p \in P} g(y_p, \eta_p^O) \\ &= \sum\nolimits_{p \in P} \left(C_p^H (y_p - \eta_p^O)^+ + C_p^L (\eta_p^O - y_p)^+ \right) \end{aligned} \tag{13}$$

where $\mathrm{g}(y_p, \eta_p^O)$ represents the EC holding and leasing cost of port p in one period; $x^+ = max\ (0, x)$.

Next, we consider the problem under balanced scenario, followed by that under unbalanced scenario. Here, balanced (unbalanced) scenario is the scenario in which the total amount of estimated EC supply is equal (not equal) to the total amount of estimated EC demand in each period. From (3) and (4), it is observed that $\sum_{i \in P_t^S} a_{i,t}^S - \sum_{j \in P_t^D} a_{j,t}^D = N - \sum_{p \in P} \gamma_p$. Hence, a scenario with $\sum_{p \in P} \gamma_p = N$ is a balanced scenario and a scenario with $\sum_{p \in P} \gamma_p \neq N$ is an unbalanced scenario.

1) Balanced Scenario: Consider the problem in the balanced scenario. We can obtain the optimal solution of (10) analytically, since it only depends on the holding and leasing cost function. The explanations are as follows.

From the transportation models, we know that the repositioning out (in) requirement of each surplus (deficit) port can be fully satisfied in each period in a balanced scenario. Thus, after making ECR decisions, the inventory position level of each port can be always kept at its target threshold level. It implies that the estimated EC supply (demand) of the surplus (deficit) port in a period, except the initial period, will be independent from the parameters of the fleet size and the thresholds, and just depend on its customer demands in the previous period. Consequently, the EC transportation cost and the total EC holding and leasing cost in one period will be independent; and the expected EC transportation cost per period will be independent from parameters of N and $\boldsymbol{\gamma}$. Further speaking, the optimal solution only depends on the expected total EC holding and leasing cost function. The problem (10) in the balanced scenario can be simplified to an non-linear programming as:

$$\min_{\boldsymbol{\gamma}} \ E\left(\sum\nolimits_{p \in P} \left(C_p^H (\gamma_p - \eta_p^O)^+ + C_p^L (\eta_p^O - \gamma_p)^+ \right) \right)$$

$$\text{s.t. } \sum\nolimits_{p \in P} \gamma_p = N$$

where the value of fleet size N is given.

Considering the convexity of the above cost function and taking use of the K.K.T. conditions, we can obtain the optimal solution of the NLP by solving (14) and (15).

$$\left(C_p^H + C_p^L \right) \bullet F_p(\gamma_p) - C_p^L + \lambda_N = 0 \ \ \forall p \in P \tag{14}$$

$$\sum\nolimits_{p \in P} \gamma_p - N = 0 \tag{15}$$

where λ_N is the Lagrange Multiplier of the balance constraint.

Remark: Let $(\boldsymbol{\gamma}^*)^B$ be the optimal thresholds in the balanced scenario with given fleet size. Given customer demands, we know that $(\boldsymbol{\gamma}^*)^B$ can achieve the minimum holding and leasing cost for the problem (10). However, it may not achieve the minimum expected total cost per period for the problem, because we can find other thresholds achieving less transportation costs than that achieved by $(\boldsymbol{\gamma}^*)^B$. For example, when we set all the thresholds going to infinite so that no ECs will be repositioned, the transportation cost in this scenario will be zero and less than that in the scenario with $(\boldsymbol{\gamma}^*)^B$. Thus, we next consider the unbalanced scenario.

2) Unbalanced Scenario: Consider the problem in the unbalanced scenario. By taking advantage of the structure of the problem, we find an interesting property about the transportation cost as follows:

Property I: *In an unbalanced scenario, the transportation cost in a period, except the initial period, could be less than or equal to that in a balanced scenario with same customer demands.*

Intuitively, for example, if a exported-dominated port has the probability to become a surplus port, i.e., it needs to reposition out ECs to other ports, repositioning in less ECs than its threshold in this port in advance when it becomes a surplus port will reduce its EC repositioning out quantity. Hence, less transportation cost in a period in an unbalanced scenario could be occurred.

Since there is no closed-form formulation for the computation of expected total cost per period in the unbalanced scenario involving the repositioned EC quantities from the transportation models, we adopt the simulation to estimate $J(N, \boldsymbol{\gamma})$ given values of N and $\boldsymbol{\gamma}$ as shown in (16).

$$J\left(N, \boldsymbol{\gamma}\right) \approx \frac{1}{T} \sum\nolimits_{t=1}^{T} J\left(\mathbf{x}_t, \boldsymbol{\gamma}, \omega_t\right) = \frac{1}{T} \sum\nolimits_{t=1}^{T} \left(H\left(\mathbf{x}_t, \boldsymbol{\gamma}\right) + G\left(\mathbf{y}_t, \omega_t\right) \right) \tag{16}$$

where $J(\mathbf{x}_t, \boldsymbol{\gamma}, \omega_t)$ is the total cost in period t; $H(\mathbf{x}_t, \boldsymbol{\gamma})$ and $G(\mathbf{y}_t, \omega_t)$ are the EC repositioning cost and the total EC

ISBN: 978-1-61208-161-8

holding and leasing cost in period t and can be obtained from (12) and (13), respectively; T is the amount of the simulation periods. It is significant to highlight that solving (10) in the unbalanced scenario is difficult. In order to find an optimal solution to the problem, we need to use a search-based method. In next section, we develop an optimization technique namely IPA-based gradient technique.

Summarizing above discussions, we can get that given fleet size and customer demands, the minimum expected total cost per period could be achieved in either the balanced scenario or unbalanced scenario. The optimal solution under balanced scenario can be obtained analytically by solving (14) and (15), and under unbalanced scenario by applying the proposed IPA-based gradient technique.

III. IPA-BASED GRADIENT TECHNIQUE

IPA is able to estimate the gradient of the objective function from one single simulation run, thus reducing the computational time. Moreover, it has been shown that variance of IPA estimator is lower, compared with many other gradient estimators [15]. Thus, we propose an IPA-based gradient technique to search the optimal solution in the unbalanced scenario. The overall IPA-based gradient technique is briefly described in Fig. 1.

As shown in Fig. 1, given the parameters of the fleet size N and the policy $\boldsymbol{\gamma}$ with $\sum_{p\in P}\gamma_p \neq N$, we first calculate the total cost and estimate the gradient of total cost with respect to the thresholds in all periods. We estimate the gradient in a period using the concept of perturbation propagation from IPA [16], the dual information of the LP model and the chain rule. Then, we can obtain the expected total cost per period and the gradient of $J(N,\boldsymbol{\gamma})$. This gradient can provide a direction for finding new parameters of the policy that may have a lower expected total cost per period and hence the hill climbing algorithm is used to update the parameters of the policy. Finally, when the termination criteria are satisfied, the simulation is stopped.

To estimate the gradient of expected total cost per period, we take a partial derivation of (16) with respect to the threshold of port i. With the help of (13), we can obtain

$$\begin{aligned}\frac{\partial J(N,\boldsymbol{\gamma})}{\partial\gamma_i} &\approx \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial J\left(\mathbf{x}_t,\boldsymbol{\gamma},\omega_t\right)}{\partial\gamma_i}\right)\\ &=\frac{1}{T}\sum_{t=1}^{T}\left(\frac{\partial H(\mathbf{Z}_t)}{\partial\gamma_i}+\sum_{p\in P}\frac{\partial E\left(g(y_{p,t},\eta_{p,t}^{O})\right)}{\partial y_{p,t}}\bullet\frac{\partial y_{p,t}}{\partial\gamma_i}\right)\end{aligned} \tag{17}$$

where for the inventory holding and leasing cost function, we use the expected holding and leasing cost function to estimate the gradient instead of using the sample path since we are able to get the explicit function to evaluate the average gradient; $\partial H(\mathbf{Z}_t)/\partial\gamma_i$ measures the impact of the transportation cost in period t when the threshold is changed; $\partial g(y_{p,t},\eta_{p,t}^{O})/\partial y_{p,t}$ measures the impact of the holding and leasing cost function of port p in period t when the inventory position level is changed; $\partial y_{p,t}/\partial\gamma_i$ measures the impact of the inventory position level of port p in period t when the threshold is changed.

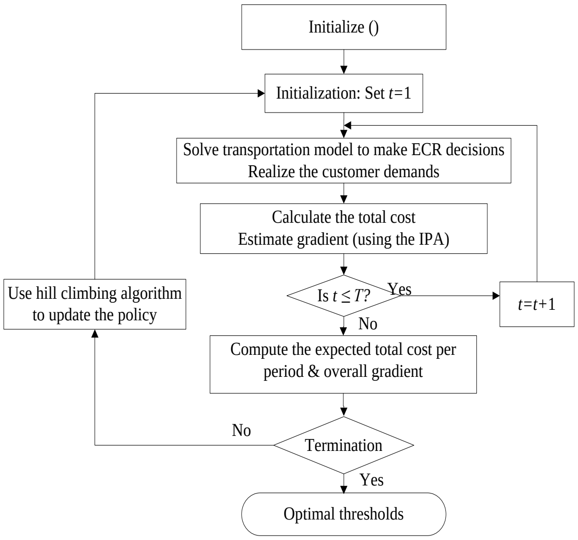

Initialize ()
Initialization: Set t=1
Solve transportation model to make ECR decisions
Realize the customer demands
Calculate the total cost
Estimate gradient (using the IPA)
Is $t \le T$?
Yes
No
t=t+1
Use hill climbing algorithm to update the policy
Compute the expected total cost per period & overall gradient
Termination
No
Yes
Optimal thresholds

Figure 1. The flow of the IPA-based gradient technique

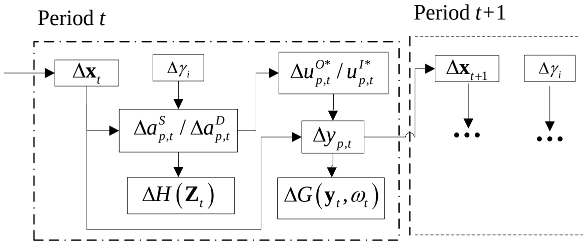


Figure 2. The Perturbation Flow

We define the nominal path as the sample path generated by the simulation model with parameter $\boldsymbol{\gamma}$ and the perturbed path as the sample path generated using the same model and same random seeds, but with parameter $(\boldsymbol{\gamma})'$, where $(\boldsymbol{\gamma})' = \boldsymbol{\gamma}+\Delta\boldsymbol{\gamma}$. Without loss of generality, we only perturb the threshold of port i and keep the thresholds of the other ports unchanged, i.e., $(\gamma_i)' = \gamma_i + \Delta\gamma_i$ and $(\gamma_j)' = \gamma_j$ for other port j, where the value of $\Delta\gamma_i$ is infinitesimally small. By "sufficiently" small, we mean such that the surplus port subset and deficit port subset are same in the both nominal and perturbed paths in every period. Oftentimes, we will present the changes in various quantities by displaying with argument Δ. We perturb γ_i in all periods and the representative perturbation flow in period t is shown in the Fig. 2.

In our problem with real variables, the probability of having balanced port is close to 0. In other word, port i should be either surplus port or deficit port in period t. From (3) and (4), we can derive that $\Delta a_{p,t}^{S} = \Delta x_{p,t} - \Delta\gamma_p\ \forall p \in P_t^S$ and $\Delta a_{p,t}^{D} = \Delta\gamma_p - \Delta x_{p,t}\ \forall p \in P_t^D$. Hence, in Fig. 2, the perturbation of $\Delta\mathbf{x}_t$ will work together with the perturbation of $\Delta\gamma_i$ to affect the estimated EC supply/demand, namely $\Delta a_{p,t}^{S}/\Delta a_{p,t}^{D}$ for some ports. We know that the estimated EC supply/demand of a port is the RHS of the corresponding port's constraint in the transportation model. It implies that the perturbations of

$\Delta a_{p,t}^{S}/\Delta a_{p,t}^{D}$ of some ports could affect the optimal repositioning quantities of some ports, which, of course, will affect the total optimal repositioned out/in quantities of some ports, namely $\Delta u_{p,t}^{O*}$ /$\Delta u_{p,t}^{I*}$. From (1), we know that $\Delta y_{p,t} = \Delta x_{p,t} - u_{p,t}^{O} + u_{p,t}^{i}\ \forall p \in P$. The perturbation of $\Delta \mathbf{x}_t$ will work together with the perturbation of $\Delta u_{p,t}^{O*}$ /$\Delta u_{p,t}^{I*}$of some ports to affect the perturbation on EC inventory positions, namely $\Delta y_{p,t}$ for some ports. Furthermore, the perturbation of $\Delta a_{p,t}^{S}/\Delta a_{p,t}^{D}$ will affect the transportation cost and the perturbation of $\Delta y_{p,t}$ will affect the total holding and leasing cost. From (2), we have $\Delta \mathbf{x}_t = \Delta \mathbf{y}_t$, which implies that the perturbation on the inventory position will be fully propagated to the beginning on-hand inventory of next period.

The flowing notations are introduced.

$Q_{p,t}$ the set of ports whose beginning on-hand inventory in period t are affected by perturbing threshold of port p, $Q_{p,t} \subset P$

$E_{p,t}$ the set of ports whose total optimal repositioned quantities are changed by perturbing the estimated EC supply/demand of port p in period t, $E_{p,t} \subset P$

$\pi_{p,t}$ the corresponding dual variable for port p constraint in the transportation model in period t

$I\{\text{condition}\}$ a indicator function, which takes 1 if the condition is true and otherwise 0

Tracing the perturbations by following the flow in Fig. 2, we can obtain that in period t, $E_{i,t}$ will be either empty or consist of a pair of ports, i.e., $E_{i,t} = \emptyset$ or $E_{i,t} = (i,\ e_{i,t})$. Similarly, $Q_{i,t}$ will be either empty or consist of a pair of ports, i.e., $Q_{i,t} = \emptyset$ or $Q_{i,t} = (i,\ q_{i,t})$. We can obtain the gradient of expected total cost per period with respect to γ_i in (17) can be approximated by (18). In (18), the first term of the RHS presents the perturbation on the transportation cost when $Q_{i,t} = \emptyset$; the second term of the RHS presents the perturbation on the transportation cost when $Q_{i,t} \neq \emptyset$; the third term of the RHS presents the perturbation on the holding and leasing cost when $Q_{i,t} = \emptyset$ and $E_{i,t} \neq \emptyset$; the forth term of the RHS presents the perturbation on the holding and leasing cost when $Q_{i,t} \neq \emptyset$ and $E_{i,t} = \emptyset$; the fifth term of the RHS presents the perturbation on the holding and leasing cost when $Q_{i,t} \neq \emptyset$, $E_{i,t} \neq \emptyset$ and $e_{q_{i,t},t} \neq i$; $E_{i,t}$ can be obtained by applying a proposed modified stepping stone approach with perturbing the estimated supply/demand of port i in period t. We know that $Q_{i,1} = \emptyset$ in the initial period; and for $t > 1$, $Q_{i,t}$ can be obtained as follows: (a) $Q_{i,t} = E_{i,t}$, when $Q_{i,t-1} = \emptyset$; (b) $Q_{i,t} = Q_{i,t-1}$, when $Q_{i,t-1} \neq \emptyset$ and $E_{q_{i,t-1},t} = \emptyset$; (c) $Q_{i,t} = \emptyset$, when $Q_{i,t-1} \neq \emptyset$, $E_{q_{i,t-1},t} \neq \emptyset$ and $e_{q_{i,t-1},t} = i$; (d) $Q_{i,t} = \left(i, e_{q_{i,t-1},t}\right)$, when $Q_{i,t-1} \neq \emptyset$, $E_{q_{i,t-1},t} \neq \emptyset$ and $e_{q_{i,t-1},t} \neq i$.

$$\frac{\partial J(N,\boldsymbol{\gamma})}{\partial \gamma_i} \approx \frac{1}{T}\sum_{t=1}^{T} \frac{\partial J(\mathbf{x}_t,\boldsymbol{\gamma},\omega_t)}{\partial \gamma_i}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \left(\begin{array}{l} I(Q_{i,t} = \emptyset) \bullet (-1)^{I(i \in P_t^S)} \bullet \pi_{i,t} + I(Q_{i,t} \neq \emptyset) \bullet (-1)^{I(q_{i,t} \in P_t^S)} \bullet \pi_{q_{i,t},t} \\ +I\begin{pmatrix} Q_{i,t} = \emptyset, \\ E_{i,t} \neq \emptyset \end{pmatrix} \bullet \left(\frac{\partial E\left(g\left(y_{i,t},\eta_{i,t}^{O}\right)\right)}{\partial y_{i,t}} - \frac{\partial E\left(g\left(y_{e_{i,t},t},\eta_{e_{i,t},t}^{O}\right)\right)}{\partial y_{e_{i,t},t}} \right) \\ +I\begin{pmatrix} Q_{i,t} \neq \emptyset, \\ E_{q_{i,t},t} = \emptyset \end{pmatrix} \bullet \left(\frac{\partial E\left(g\left(y_{i,t},\eta_{i,t}^{O}\right)\right)}{\partial y_{i,t}} - \frac{\partial E\left(g\left(y_{q_{i,t},t},\eta_{q_{i,t},t}^{O}\right)\right)}{\partial y_{q_{i,t},t}} \right) \\ +I\begin{pmatrix} Q_{i,t} \neq \emptyset, \\ E_{q_{i,t},t} \neq \emptyset, \\ e_{q_{i,t},t} \neq i \end{pmatrix} \bullet \left(\frac{\partial E\left(g\left(y_{i,t},\eta_{i,t}^{O}\right)\right)}{\partial y_{i,t}} - \frac{\partial E\left(g\left(y_{e_{q_{i,t},t},t},\eta_{q_{i,t},t,t}^{O}\right)\right)}{\partial y_{e_{q_{i,t},t},t}} \right) \end{array} \right) \tag{18}$$

where the value of $\partial E(.)/\partial(.)$ in (18) is calculated by

$$\frac{\partial E\left(g(y_{p,t},\eta_{p,t}^{O})\right)}{\partial y_{p,t}} = \begin{cases} -C_p^L & \text{if } y_{p,t} < 0 \\ \left(C_p^H + C_p^L\right) \bullet F_p\left(y_{p,t}\right) - C_p^L & \text{if } 0 \leq y_{p,t} \end{cases} \tag{19}$$

IV. NUMERICAL RESULTS

In this section we aim to evaluate the performance of the proposed single-level threshold policy (STP). For comparison, a match back policy (MBP) is introduced. Such policy is widely accepted and applied in practice and its basic principle is to match the containers back to the original port. Mathematically,

$$z_{p,m,t+1} = \left(\varepsilon_{m,p,t} - \varepsilon_{p,m,t}\right)^{+} \tag{20}$$

The NLP in the balanced scenario is solved by Matlab (version 7.0.1). The IPA-gradient based algorithm is coded in Visual C++ 5.0. All the numerical studies are tested on and Intel Duo Processor E6750 2.67GHz CPU with 4.00 GB RAM under the Microsoft Vista Operation System. We set the simulation period $T = 10{,}100$ with warm-up period T_0 =100. For the STP, the termination criteria are that the maximum iteration for finding the optimal thresholds, namely r_{max} is achieve, or the expected total cost in the iteration r is larger than that in the previous iteration. We set $r_{max} = 1{,}000$. For the MBP, since the transportation cost is independent from the parameter of fleet size, we set the inventory position in the initial period be equal to the optimal inventory position which minimizing the expected holding and leasing cost.

For a three-port system, we compare the performance of both policies based on the expected total cost per period. we give the fleet size from 483 TEUs to 1128 TEUs to investigate the effect of the fleet size on the expected total cost. Fig. 3 shows the results. It is observed that STP outperforms MBP for all cases. The expected total cost per period savings achieved by STP over MBP are of the order of 12.75%~37.18%. One possible explanation is that STP makes the ECR decisions in terms of minimizing the transportation cost. Hence, it is important for operators to use intelligent method in repositioning ECs, instead of resorting to simple way such as the MBP.

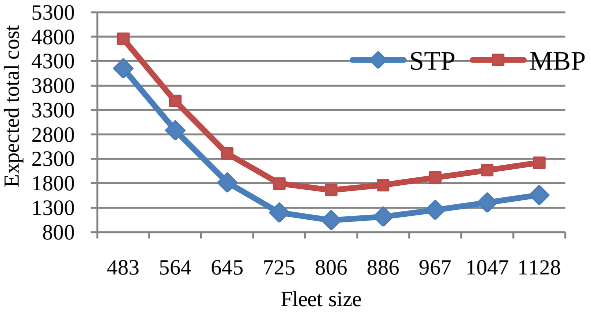


Figure 3. Expected total cost per period comparison for three-port system

From Fig. 3, it reveals that the optimal average total cost appears to be convex with respect to the fleet size for each system. It reflects the intuition that the optimal fleet size is the trade-off between the transportation cost and the holding and leasing cost.

V. Conclusion And Future Work

In this paper, the EC repositioning problem in a multi-port system is considered. A single-level inventory-based policy with the repositioning rule in terms of minimizing transportation cost is developed to reposition ECs periodically by taking into account demand uncertainty and dynamic operations. Two approaches, non-linear programming and IPA-based gradient technique are developed to solve the problem optimizing thresholds of policy under balanced and unbalanced scenarios, respectively. The numerical results provide insights that by repositioning the ECs intelligently, we can significantly reduce the total operation cost.

The main contributions of the study are as follows: (a) a single-level threshold policy with a repositioning rule in terms of minimizing transportation cost is developed for repositioning ECs in a multi-port system. To the best of our knowledge, few works consider the repositioning rule which is related to the transportation costs; (b) by developing the method to solve the difficult ECR problem, i.e., using IPA to estimate the gradient, it is innovative and provides a potential methodology contribution in this field

We strongly assumed that the ECs are dispatched between each pair of ports in one period. It may not be the right one period in some general cases. Further research is needed to relax the one-period assumption and consider the problem with different time dimension for the repositioning time. The main challenge is to track the perturbations along the sample path.

References

[1] U. Nations, Regional Shipping and Port Development (Container Traffic Forecast). 2007.

[2] D. P. Song, “Optimal threshold control of empty vehicle redistribution in two depot service systems,” IEEE Transactions on Automatic Control, vol. 50(1), Jan. 2005, pp. 87-90, doi: 10.1109/TAC.2004.841134(410) 50.

[3] P. Dejax and T. Crainic, “Survey Paper--A Review of Empty Flows and Fleet Management Models in Freight Transportation,” Transportation Science, vol. 21(4), Nov. 1987, pp. 227-248, doi: 10.1287/trsc.21.4.227.

[4] T. Crainic, M. Gendreau, and P. Dejax, “Dynamic and stochastic models for the allocation of empty containers,” Operations Research, vol. 41(1), Jan. - Feb. 1993, pp. 102-126.

[5] W. Shen and C. Khoong, “A DSS for empty container distribution planning,” Decision Support Systems, vol. 15(1), Sep. 1995, pp. 75-82, doi: 10.1016/0167-9236(94)00037-S.

[6] R. Cheung and C. Chen, “A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem,” Transportation Science, vol. 32(2), May 1998, pp. 142-162, doi: 10.1287/trsc.32.2.142.

[7] S. W. Lam, L. H. Lee, and L. C. Tang, “An approximate dynamic programming approach for the empty container allocation problem,” Transportation Research Part C, vol. 15(4), Aug. 2007, pp. 265-277, doi: 10.1016/j.trc.2007.04.005.

[8] S. Choong, M. Cole, and E. Kutanoglu, “Empty container management for intermodal transportation networks,” Transportation Research Part E, vol. 38(6), Nov. 2002, pp. 423-438, doi: 10.1016/S1366-5545(02)00018-2.

[9] J. X. Dong and D. P. Song, “Container fleet sizing and empty repositioning in liner shipping systems,” Transportation Research Part E, vol. 45(6), Nov. 2009, pp. 860-877, doi: 10.1016/j.tre.2009.05.001.

[10] D. P. Song, “Characterizing optimal empty container reposition policy in periodic-review shuttle service systems,” Journal of the Operational Research Society, vol. 58(1), 2007, pp. 122-133, doi: 10.1057/palgrave.jors.2602150.

[11] J. A. Li, K. Liu, S. C. H. Leung, and K. K. Lai “Empty container management in a port with long-run average criterion,” Mathematical and Computer Modeling, vol. 40(1-2), July 2004, pp. 85-100, doi: 10.1016/j.mcm.2003.12.005.

[12] D. P. Song and J. Carter, “Optimal empty vehicle redistribution for hub-and-spoke transportation systems,” Naval Research Logistics, vol. 55(2), Mar. 2008, pp. 156-171, doi: 10.1002/nav.20274.

[13] D. P. Song and J. X. Dong, “Empty Container Management in Cyclic Shipping Routes,” Maritime Economics & Logistics, vol. 10(4), Dec. 2008, pp. 335-361, doi: 10.1057/mel.2008.11.

[14] J. A. Li, S. C. H. Leung, Y. Wu, and K. Liu, “Allocation of empty containers between multi-ports,” European Journal of Operational Research, vol. 182(1), Oct. 2007, pp. 400-412, doi: 10.1016/j.ejor.2006.09.003.

[15] R. Suri, “Perturbation analysis: the state of the art and research issuesexplained via the GI/G/1 queue”, Proceedings of the IEEE, vol. 77(1), Jan. 1989, pp. 114-137, doi: 10.1109/5.21075.

[16] Y.C.Ho and X. R. Cao, Discrete Event Dynamic Systems and Perturbation Analysis. Boston,UK, Kluwer Academic Publishers, 1991.

Sample Average Approximation for Stochastic Empty Container Repositioning

Ek Peng CHEW[a], Loo Hay LEE[b], Yin LONG[c]
Department of Industrial & Systems Engineering
National University of Singapore
Singapore 119260
e-mail: [a]isecep@nus.edu.sg; [b] iseleelh@nus.edu.sg; [c]iselongy@nus.edu.sg

***Abstract*—To incorporate uncertainties in empty container repositioning problem, we formulate a two-stage stochastic programming model with random demand, supply, ship weight capacity and ship space capacity. The Sample Average Approximation (SAA) method is applied to approximate the expected value function. Several non-independent and identically distributed sampling schemes are considered to enhance the performance of the SAA method. Numerical experiments show that near-optimal solutions could be provided by the SAA method with these sampling schemes.**

Keywords-empty container repositioning; simulation optimization; sample average approximation; supersaturated design

I. INTRODUCTION

One main issue in the containerized transportation is the imbalance of container flow, which is the result of global trade imbalance between different regions. Thus, maintaining higher operational cost efficiencies in repositioning empty containers becomes a crucial issue in shipping industry.

There are increasing studies on empty container flows in recent years [1][2]. In the maritime transportation, container operators have to deal with some uncertain factors like the real transportation time between two ports/deports, future demand and supply, the in-transit time of returning empty container from customers, and the available capacity in vessels for empty containers transportation, etc. The uncertain nature of parameters is taken into account in several studies [3][4]. A multi-scenario model was proposed in [5] to address the Empty Container Repositioning (ECR) problem in a scheduled maritime system. In their study, opinions of shipping companies were considered to generate scenarios when the distributions of uncertain parameters cannot be estimated through historical data. Our study focuses on developing scenario-based model when the distribution of uncertain parameters can be estimated through historical data. Random scenarios could be generated based on these distributions. However, it is difficult to solve the stochastic ECR problem with a large number of scenarios. In this case, we apply the Sample Average Approximation (SAA) method to solve the stochastic ECR problem with multiple scenarios.

The SAA problem with multiple scenarios is usually difficult to solve due to its large scale. In this study, we try to enhance the SAA method with well-planned samplings which are more representative. The motivation of this idea is to get acceptable solutions by solving SAA problems with a small number of scenarios. Independent and identically distributed (i.i.d) sampling is well studied for construction approximations [6][7]. On the other hand, SAA with non-i.i.d samplings is also studied in recent year [8][9]. Empirically, it was shown that Latin Hypercube (LH) and Antithetic Variates (AV) methods outperform those under i.i.d sampling, with LH outperforming AV [10]. In [11], U design was used to further enhance the accuracy of the SAA and their theoretical results showed that the SAA with U designs can significantly outperform those with LH designs. In this study, we try to do as a whole very few experiments (even less than the number of degrees of freedom of the system when that is possible) and still get a satisfying approximation. To our knowledge, no existing study applies the SAA method with non-i.i.d samplings to the stochastic ECR problem where the distributions of uncertain parameters are known. This study is to fill in this gap.

The paper is structured in the following way. Section I concerns introduction; In Section II, we provide the description of a basic deterministic model and our two-stage stochastic model for ECR. Section III shows the solving methodologies to solve our proposed model. The SAA method and the sampling schemes to enhance SAA are explained. Section IV presents the results of computational studies. Finally, we give conclusions and outline directions for future research in Section V.

II. PROBLEM FORMULATION

The focus of this study is to make operational level maritime ECR decisions for shipping companies. As the global container transportation network is large and complex, the ocean liners usually divide the global network into several regions and appoint regional operators to manage the container flows for each region. Because of the long lead time of the across-region empty containers, ocean liners usually make ordering decisions depending on forecasting to make decisions on ordering. Due to the booking system used in the maritime transportation, demand, supply and the available ship capacity in the near future could be forecasted accurately. However, it is difficult to obtain accurate forecasting for more than one or two weeks. Currently, container operators make decisions based on the nominal forecast value. Because of the differences between the expected value and the realized value, inefficient solutions

ISBN: 978-1-61208-161-8

may be produced. These decisions have to be recovered at real-time operations.

A. *The Deterministic Model*

If all parameters in the planning horizon are known, the deterministic time space model for ECR could be formulated as follows, where the actual service schedule and most port requirements are considered. Details of this model could be found in [12].

$$\text{Min} \sum_{t=1,2,\dots,\mathrm{T}} TC^t$$

$$\begin{aligned} TC^t = & \sum_{k\in K}\sum_{i\in P} c^u_{t,i,k} \sum_{(s,v)\in\{(s,v)|v\in V, s\in S_v, p_{v,s}=i, t\in A_{v,s}\}} u_{t,s,v,k} \\ & + \sum_{k\in K}\sum_{i\in P} c^w_{t,i,k} \sum_{(s,v)\in\{(s,v)|v\in V, s\in S_v, p_{v,s}=i, t\in D_{v,s}\}} w_{t,s,v,k} \\ & + \sum_{i\in P}\sum_{k\in K} (c^y_{t,i,k} y_{t,i,k} + c^z_{t,i,k} z_{t,i,k}) \\ & + \sum_{k\in K} \sum_{(s,v)\in\{(s,v)|v\in V, s\in S_v, t\in D_{v,s}\}} c^x_{t,s,v,k} x_{t,s,v,k} \end{aligned}$$

Subject to

$$\sum_{k\in K} (g_k \times x_{t,s,v,k}) \le \gamma_{t,s,v} \quad \forall\ (v,s,t) \in \{(v,s,t)|v\in V, s\in S_v, t\in D_{v,s}\} \qquad (1)$$

$$\sum_{k\in K} (h_k \times x_{t,s,v,k}) \le \sigma_{t,s,v} \quad \forall\ (v,s,t) \in \{(v,s,t)|v\in V, s\in S_v, t\in D_{v,s}\} \qquad (2)$$

$$\begin{aligned} & x_{t-b_{v,s},s,v,k} - u_{t,s+1,v,k} + w_{t+d_{v,s},s+1,v,k} = x_{t+d_{v,s},s+1,v,k} \\ & \qquad \forall\ (v,s,t) \in \{(v,s,t)|v\in V, s\in S_v, t\in D_{v,s}\} \end{aligned} \qquad (3)$$

$$\begin{aligned} & x_{t-b_{v,s},s,v,k} \ge u_{t,s+1,v,k} \\ & \qquad \forall k\in K, \forall (v,s,t) \in \{(v,s,t)|v\in V, s\in S_v, t\in A_{v,s+1}\} \end{aligned} \qquad (4)$$

$$\begin{aligned} & y_{t-1,i,k} + \sum_{(s,v)\in\{(s,v)|v\in V, s\in S_v, p_{v,s}=i, t\in A_{v,s}\}} u_{t,s,v,k} - \psi_{t,i,k} + z_{t,i,k} + \theta_{t,i,k} \\ & - \sum_{(s,v)\in\{(s,v)|v\in V, s\in S_v, p_{v,s}=i, t\in D_{v,s}\}} w_{t,s,v,k} = y_{t,i,k} \\ & \qquad \forall\ k\in K, \forall i\in P, \forall t = 1,2,\dots,T \end{aligned} \qquad (5)$$

$$u_{t,s,v,k}, w_{t,s,v,k}, x_{t,s,v,k}, y_{t,i,k}, z_{t,i,k} \ge 0 \qquad (6)$$

The objective function is to minimize the total operation cost in the planning horizon. TC^t is the total operational cost at time *t*. The operational costs include the handling cost (unloading cost and loading cost), the holding cost, the penalty cost and the transportation cost. $u_{t,s,v,k}$, $w_{t,s,v,k}$, $x_{t,s,v,k}$, $y_{t,i,k}$, and $z_{t,i,k}$ are decision variables, where $u_{t,s,v,k}$ is the number of empty containers of size k unloaded at stop *s* from service *v* at time *t*, $w_{t,s,v,k}$ is the number of empty containers of size k loaded from stop *s* onto service *v* at time *t*, $x_{t,s,v,k}$ is the number of empty containers of size *k* transported from stop *s* to next stop on service *v* leaving stop *s* at time *t*, $y_{t,i,k}$ is the number of empty containers of size *k* stored at port *i* at time *t*, and $z_{t,i,k}$ is the number of empty containers of size *k* that cannot be satisfied by the empty containers stored at port *i* at time *t*. $c^u_{t,i,k}$, $c^w_{t,i,k}$, $c^x_{t,s,v,k}$, $c^y_{t,i,k}$, and $c^z_{t,i,k}$ are the corresponding cost parameters of unloading, loading, transportation, storing and penalty respectively. *V* is the set of services. *P* is the set of ports. *Q* is the set of regions. *K* is the set of container sizes. And S_v is the set of stops on service *v*. $D_{v,s}$ is the set of periods in which service *v* departs from its stop *s*. $A_{v,s}$ is the set of periods in which service *v* arrives at its stop *s*.

Constraint (1) and constraint (2) are ship capacity constraints. $\gamma_{t,s,v}$ is residual space capacity on service *v* when it leaves stop *s* at time *t*, and $\sigma_{t,s,v}$ is residual weight capacity on service *v* when it leaves stop *s* at time *t*. g_k and h_k are volume and weight of one container of size *k*. These two constraints should be considered when there is a service leaving the port. Constraint (3) guarantees the balance of the container flows at each service. $b_{v,s}$ is the transportation time from stop *s* to next stop on the service *v*. $d_{v,s}$ is the number of days that the service *v* stays at stop *s*. Constraint (4) ensures that the number of empty containers unloaded from a vessel should not exceed the total number of empty containers in the vessel. These two constraints should be considered when there is a service arriving at a port. Constraint (5) considers the balance of the container flows at each port at each time. $\theta_{t,i,k}$ is the supply of empty containers of size *k* in port *i* at time *t*, and $\psi_{t,i,k}$ is the demand of empty containers of size *k* in port *i* at time *t*. $p_{v,s}$ is the port corresponding to the stop *s* on service *v*. Constraint (6) ensures that all variables are non-negative.

B. *The Stochastic Model*

In this study, we develop a stochastic programming model which takes account into four uncertain parameters, i.e. the demand (the empty containers that picked up by the customers to load cargos), the supply (the empty containers that returned by the customers), the available ship space capacity and available ship weight capacity for empty containers. Other uncertain factors like the transportation time between two ports are not considered. We also do not consider container substitution in this study. We assume that service schedule is given and fixed in the planning horizon. This assumption is valid as the planning horizon of our operation model is short (several weeks), and the service schedule is not changed frequently. In order to incorporate the deterministic information and the uncertain information, a two-stage stochastic programming is developed. This model is run in a rolling horizon manner. ECR decisions are made at the beginning of stage 1 and will be made again when new information is collected.

Let $\omega \in \Omega$ denotes a scenario that is unknown when decisions at stage 1 are made, but that is known when the decisions at stage 2 are made, where Ω is the set of all scenarios. A two-stage stochastic model for ECR is formulated as follows.

Stage 1

$$\min \quad g(x_1) = c_1 x_1 + E_p[Q(x_1, \xi(\omega))] \tag{7}$$

$$\text{subject to} \quad A_1 x_1 = a_1 \tag{8}$$

$$B_1 x_1 = v \tag{9}$$

$$x_1 \geq 0 \tag{10}$$

Stage 2: For a realized scenario ω, we have

$$Q[x_1, \xi(\omega)] = \min \quad c_2\, x_2(\omega) \tag{11}$$

$$\text{subject to} \quad A_2 x_2(\omega) = a_2(\omega) \tag{12}$$

$$B_2 x_2(\omega) = v(x_1) \tag{13}$$

$$x_2(\omega) \geq 0 \tag{14}$$

x_1: Decisions at stage 1

x_2 (ω): Decisions for scenario ω at stage 2 given x_1

$c_{1,}$ c_2 : The cost vector at stage 1 and stage 2 respectively

A_1, B_1, A_2, B_2: The coefficient matrices of x_1 or x_2

$a_{1,}$ $a_2(\omega)$: The RHS of constraint (8) and constraint (12) respectively

v: The vector of end container states of stage 1. It is the empty container inventory at each port and at each vessel at the end of stage 1. $v = \{v_1, v_2, ..., v_K\}$

$v(x_1)$: The vector of initial container states of stage 2 given x_1

The objective function of stage 1 is to minimize the total operational cost in the planning horizon. $c_1 x_1$ is the operation cost at stage 1. $E_p[Q(x_1, \xi(\omega))]$ is the expected cost at stage 2, where p is the probability distribution of uncertain parameters. We assume that the probability distribution p on Ω is known in the stage 1. $Q(x_1, \xi(\omega))$ is the objective function of stage 2, which is the operational cost at stage 2 given x_1 and scenario ω. Constraint (8) and constraint (12) includes the typical constraints of ECR problem in the deterministic model, i.e. ship capacity constraint, service flow constraint, and port flow constraint. Constraint (9) is to set the end container states of stage 1, which are also the initial container states of stage 2. Constraint (13) is to set the initial container states of stage 2.

III. Solving Methodology

Our stochastic problem is hard to solve as it is difficult to evaluate the expected cost of stage 2 for a given x_1, i.e. $E_p[Q(x_1, \xi(\omega))]$. It requires the solutions of a large number of stage 2 optimization problems. In this study, we consider applying the SAA method to solve the stochastic ECR problem. The basic idea of SAA method is that the expected objective function of the stochastic problem is approximated by a sample average estimate derived from a random sample and the resulting SAA problem could then be solved by deterministic optimization techniques [6].

A. *The SAA Method*

A sample with N scenarios { ω^1 , ω^2 ,…, ω^N } is generated according to the probability distribution *p*. The SAA problem is formulated as follows.

$$\hat{g}_N = \min \quad c_1\, x_1 + \frac{1}{N}\sum_{n=1}^{N}[c_2 x_2(\omega^n)] \tag{15}$$

Subject to (8), (9); and (12), (13) *for n=1,2,…,N*

$$v \geq 0\,,\, x_1 \geq 0\,,\, x_2(\omega^n) \geq 0 \quad \textit{for n=1,2,…,N} \tag{16}$$

The SAA problem can then be solved by deterministic optimization methods. The optimal solution $\hat{x}$ and the optimal value $\hat{g}_N$ of the SAA problem could be obtained. $\hat{x}$ is a candidate solution of the true problem. Note that when this sample is an i.i.d random sample of the random vector, $\hat{g}_N(x)$ is called a (standard) Monte Carlo estimator of $g(x)$. To evaluate the objective value given the candidate solution $\hat{x}$, we consider generating an independent sample with *N'* scenarios, where *N'* is much larger than *N*. Let

$$\hat{g}_{_{N'}}(\hat{x}) = \min \quad c_1\, \hat{x} + \frac{1}{N'}\sum_{n=1}^{N'} Q[\hat{x}, \xi(\omega^n)] \tag{17}$$

$\hat{g}_{_{N'}}(\hat{x})$ is defined to estimate the objective value $g(\hat{x})$ of an feasible solution $\hat{x}$.

In order to get a better solution, we can generate *M* independent samples equally with *N* scenarios. By solving the corresponding *M* independent SAA problems, we can get *M* candidate solutions $\hat{x}_N^1$, $\hat{x}_N^2$, …, $\hat{x}_N^M$ and the objective values $\hat{g}_N^1$, $\hat{g}_N^2$, …, $\hat{g}_N^M$. It is natural to take $\hat{x}^*$ as one of the optimal solutions of these SAA problems which provides the smallest estimated objective value,

$$\hat{x}^* \in \arg\min\{\hat{g}_{N'}(\hat{x}_N) : \hat{x}_N \in \{\hat{x}_N^1, \hat{x}_N^2, ..., \hat{x}_N^M\}\} \tag{18}$$

To estimate the performance of SAA method, we need to calculate the optimality gap, i.e. the difference between the lower bound and the upper bound. This gap can be used to evaluate the quality of the solution. As $\hat{x}^*$ is a feasible solution of the stochastic ECR problem, $\hat{g}_{_{N'}}(\hat{x}^*)$ gives an estimate of the upper bound of the true optimal objective value of the true problem. On the other hand, as *N* realized scenarios are considered in the SAA problems, the objective value of the SAA problem $\hat{g}_N$ has a negative bias. Let $\overline{g}_N^M$ denotes the average objective value of the *M* SAA problems,

$$\overline{g}_N^M = \frac{1}{M}\sum_{m=1}^{M}\hat{g}_N^m \tag{19}$$

$\overline{g}_N^M$ provides a statistical estimate for a lower bound of the true optimal value of the true problem. The optimality gap could be estimated as

$$\hat{g}_{N'}(\hat{x}^{*}) - \bar{g}_{N}^{M} \tag{20}$$

B. Non-i.i.d Samplings

Due to its large scale, the SAA problem (15)-(16) for the real scale ECR is difficult to solve. In this case, the sampling should be well-planned. We try to generate samplings with a small number of scenarios (the number of scenarios is even less than the random variables of the stochastic ECR problem) and still get acceptable solutions. In this study, three sampling schemes are considered to enhance the performance of the SAA method for the stochastic ECR problem.

1) Latin Hypercube (LH) Sampling: In computer experiments, it is well know that LH design achieves maximum stratification in one-dimensional projections. The idea is to partition the sample space, and the number of sample points on each region should be proportional to the probability of that region. This way we ensure that the number of sampled points on each region will be approximately equal to the expected number of points to fall in that region.

2) AG Design: AG deign is a supersaturated design which is introduced in [13]. One good property of the AG design is that the saturation increase rather fast with the number of scenarios. Besides, for a two-level design with m scenarios and n factors ($m \times n$), each column has the same number of -1's and 1's in an even case. This property is necessary for a stable regression analysis as each variable has to be evaluated fairly from its smallest values to its highest values.

3) AGLH Design: We also propose a superstaturated design which combines the AG deisn and LH sampling, e.g., a two-level case, we can generate the AGLH design as follows,

a) Generate a AG design, B

b) Randomly permute the rows, columns and symbols of B ($m \times n$)

c) In each columns of B, replace the m/2 0s by a uniform random permutation of 1,…,m/2. The m/2 1s by a uniform random permutation of m/2+1,…,m.

d) Coupling B with U[0,1] random variables and we can get our desire design, C.

IV. Numerical Study

To evaluate the performance of the SAA method, we first generate an ECR transportation network as shown in Fig. 1. Five ports, three services, and one type of container (twenty-foot standard container) are considered. The planning horizon is three weeks, and we define the first week as stage 1 and the second and the third weeks are stage 2. All information in the first week is known when decisions in stage 1 are made, while some parameters in stage 2 are unknown when decisions in stage 1 are made. These parameters are known when decisions in stage 2 are made. The lead time of across-region empty containers is one week. The service schedules are given in Fig. 1.

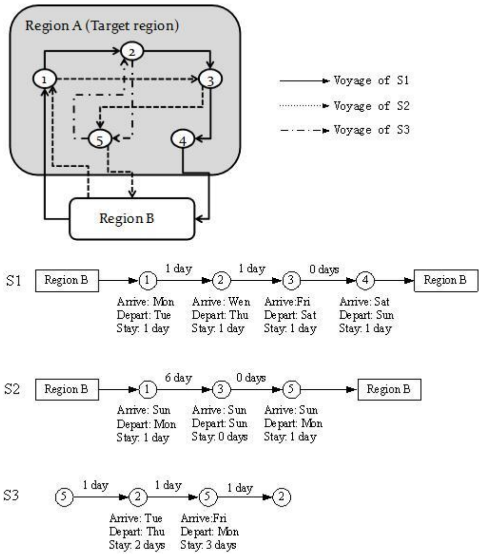


Figure 1. A network with three services and five ports

We apply the SAA method to solve the two-stage stochastic problem (with i.i.d sampling). We can solve the SAA problem directly by using CPLEX11.2 when the sample size N is not too large (N < 1000). We set the sample size N as 100. The number of scenarios to evaluate the solution N' is set to be 1000. Replication number is set to be 20. The performance of the SAA method (N = 100) for the small scale case is examined with the key results shown in Table I. As shown in Table I, the estimated objective value of the true problem $\hat{g}_{N'}(\hat{x}^{*})$ is 3359.97, and a statistic lower bound for the objective value $\bar{g}_{N}^{M}$ is 3335.45. The optimality gap $\hat{g}_{N'}(\hat{x}^{*}) - \bar{g}_{N}^{M}$ is 24.52 (0.73% of $\hat{g}_{N'}(\hat{x}^{*})$) which is quite small. The small optimality gap implies that the SAA method can provide solutions with good quality.

In the case study above, samples are independently and identically distributed. We also consider applying the supersaturated design to generate samples. The results of the SAA method with AG design are shown in Table II. The optimal estimated objective value we can obtain with N=10 (note that the number of sample scenarios is smaller that the number of random variables of the ECR problem, i.e. 56) is 3361.81, which is quite close to the optimal estimated objective value in Table I, i.e. 3359.97, with N=100. It indicates that SAA method based on supersaturated design

TABLE I. RESULTS OF THE SAA METHOD (N=100)

Estimate	Value
$\bar{g}_{N}^{M}$	3335.45
$\hat{g}_{N'}(\hat{x}^{*})$	3359.97
$\hat{g}_{N'}(\hat{x}^{*}) - \bar{g}_{N}^{M}$	24.52(0.73%)

ISBN: 978-1-61208-161-8

can provide good solutions with a small number of sample scenarios.

The performances of the SAA method with i.i.d sampling, LH sampling, AG design, and AGLH design are compared in Fig. 2 and Fig. 3 (all with *N*=10). The replication number is 1000 and we can obtain 1000 feasible solutions for each sampling method. In Fig. 2, the mean and confidence interval of the estimated objective value $\hat{g}_{N'}(\hat{x})$ of each sampling method are analyzed. We find that all the three non-i.i.d samplings can reduce the average estimated objective value and the variance of the estimated objective value. Fig. 3 is the probability plot. Based on Fig. 3, we find that the non-i.i.d samplings are less likely to provide bad solutions compared with the i.i.d sampling. We also find that the SAA method with AGLH design has the smallest probability to provide bad solutions.

TABLE II. RESULTS OF THE SAA METHOD (*N*=10, AG DESIGN)

Estimate	Value
$\bar{g}_N^M$	3275.80
$\hat{g}_{N'}(\hat{x}^*)$	3361.81
$\hat{g}_{N'}(\hat{x}^*)-\bar{g}_N^M$	86.01(2.56%)

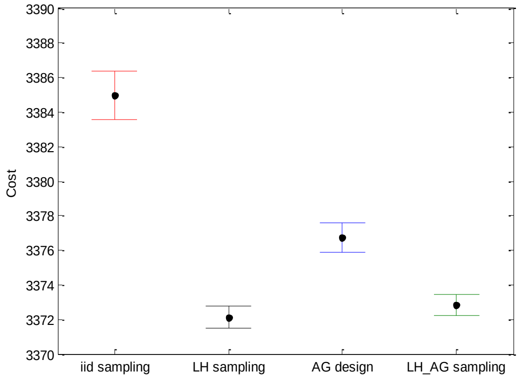


Figure 2. Expect cost estimates (α=0.05)

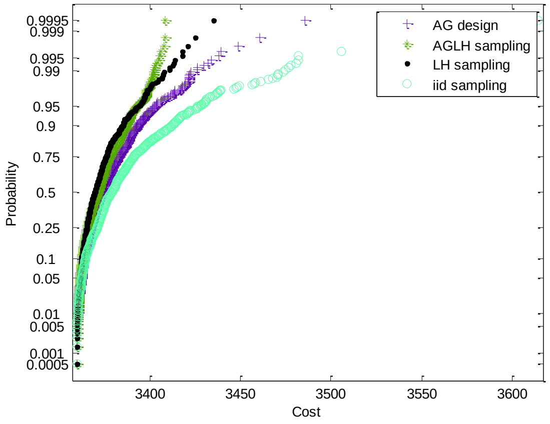


Figure 3. Probability plot of the objective estimates

V. CONCLUSION AND FUTURE STUDY

In this study, we developed an operational model to solve the ECR problem. In order to incorporate uncertainties, we built a two-stage stochastic model with uncertain demand, supply, residual ship space capacity, and residual ship weight capacity. The distributions of these parameters can be estimated based on historical data. We applied the SAA method to solve this stochastic problem. In the future, we will consider applying the SAA method to real-scale problems. Based on the results in numerical study section, we found that using LH design, AG design, and the combination of AG design and LH design to enhance the performance of SAA is promising. A direct extension of this work is to explore other sampling schemes to control scenario generation and thus improve the quality of solutions. Another possible direction for future research is to study the convergence rate of these samplings for stochastic programming.

REFERENCES

[1] A. Olivo, P. Zuddas, M. D. Francesco, and A. Manca, “An Operational Model for Empty Container Management,” Maritime Economics and Logistics, vol. 7, pp. 199-222, 2005.

[2] C. M. Feng and C. H. Chang, “Empty Container Reposition Planning for intra-Asia Liner Shipping,” Maritime Policy and Management, vol. 35, pp. 469-489, 2008.

[3] R. K. Cheung and C. Y. Chen, “A Two-Stage Stochastic Network Model and Solution Method for the Dynamic Empty Container Allocation Problem,” Transportation Science, vol. 32, pp. 142-162, 1998.

[4] A. L. Erera, J. C. Morales, and M. Savelsbergh, “Robust optimization for empty repositioning problems,” Operations Research, vol. 57, pp. 468-483, 2009.

[5] M. D. Francesco, T. G. Crainic, and P. Zuddas, “The effect of multi-scenario policies on empty container repositioning,” Transportation Research Part E: Logistics and Transportation Review, vol. 45, pp. 758-770, 2009.

[6] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello, “The Sample Average Approximation Method for Stochastic Discrete Optimization,” SIAM Journal on Optimization, vol. 12, pp. 479-502, 2002.

[7] J. Wei and M. J. Realff, “Sample average approximation methods for stochastic MINLPs,” Computers and Chemical Engineering, vol. 28, pp. 333-346, 2004.

[8] T. Homem-De-Mello, “On Rates of Convergence for Stochastic Optimization Programs Non-Independent and Identically distributed Sampling,” SIAM J. OPTIM, vol. 19, pp. 524-551, 2008.

[9] H. Xu, “Uniform Exponential Convergence of Sample average random functions under general sampling with allocation in stochastic programming,” Journal of Mathematical Analysis and Applications, vol. 368, pp. 692-710, 2010.

[10] M. B. Freimer, J. T. Linderoth, and D. Thomas, “The Impact of sampling methods on bias and variance in stochastic linear programs.” Computational Optimization and Applications. 2010, DOI: 10.1007/s10589-010-9322-x

[11] B. Q. Tang and P. Z. G. Qian, “Enhancing the sample average approximation method with U design,” Biometrika 2010, pp. 1-14.

[12] L. H. Lee, E. P. Chew, Y. Long, Y. Luo, and J. J. Shao, “A Dynamic Approach to Empty Container Flow Management,” International Association of Matitime Economists 2010 Annual Conference, Lisbon, Portugal, July, pp. 7-9.

[13] S. Ahlinder and I. Gustafsson, “On Super Saturated Experiment Design,” unpublished.

Analysis of Video Streaming Performance in Vehicular Networks

Sergi Reñé, Carlos Gañán, Juan Caubet,
Juanjo Alins, Jorge Mata-Díaz and Jose L. Muñoz
Departament d′Enginyeria Telemàtica
Universitat Politècnica de Catalunya (UPC)
1-3 Jordi Girona, C3 08034 Barcelona (Spain)
{sergi.rene, carlos.ganan, juan.caubet, juanjo, jmata, jose.munoz}@entel.upc.edu

***Abstract*—Vehicular Ad-hoc Networks (VANETs) have been mainly motivated for safety applications, but non-safety applications can also be very helpful to impulse vehicular networks. Among non-safety applications, video streaming services can provide attractive features to many applications and can attract a great number of users. However, VANETs high mobility characteristics and packet loss during communications blackouts difficult the deployment of video services in vehicular networks. In this paper, the performance of a video streaming service has been analyzed to study the deployability of a video on demand service in a highway environment for vehicular users. It has been analyzed the packet loss produced by network reconfiguration during handoffs and its influence in the video streamed quality. Using Mobile IP without and with fast handoffs we have gauge the effects of mobility over the video transmission. We show that although fast handoffs techniques minimize blackouts, they limit the deployment of video streaming services in vehicular networks.**

***Index Terms*—vehicular network emulation, mobility management, video streaming.**

I. Introduction

Mobility has changed the way people communicate. Nowadays, as Internet becomes more global, demands for mobility are not restricted to single terminals. Road and vehicle circulation systems are one of the most important infrastructures and are supporting the humans daily life. Intelligent Transportation Systems (ITS) aim to optimize the social costs of road systems and enhance their security as well as drivers comfort by allowing such services as fleet management, navigation, billing, multimedia applications, etc. Vehicular Ad-hoc Networks (VANETs) are becoming a reality mainly focused on navigation safety applications, but vehicular networks are not only useful for safety applications. Another kind of applications are also very important for the successful deployment of vehicular networks. In this way, infotainment services offer information and/or entertainment, e.g., Internet access, multiplayer games, multimedia applications, videoconference. These services can be an impulse not only for users, but also for network operators that could find infotainment applications an interesting business opportunity.

In this sense, vehicular networks are mainly impulsed in Europe by Car2Car Communication Consortium [1]. The C2C-C Consortium is an industry consortium of car manufacturers and electronics suppliers that focuses on the definition of an European standard for vehicular communication protocols. The consortium defines a C2C-C protocol stack that offers specialized functionalities and interfaces to safety-oriented applications and relies as a communication technology on a modified version of IEEE 802.11 [2]. This protocol stack is optionally placed beside a traditional TCP/IP stack (see Fig. 1), exclusively based on IPv6, which is mainly used for non-safety applications or potentially by any application that is not subject to strict delivery requirements, including Internet-based and multimedia applications. To allow vehicles to move from one network to another while maintaining the connection to the Internet, the C2C-C architecture optionally uses a Mobile IP solution [3] for host mobility or a Network Mobility (NEMO) Basic Support solution [4] for network mobility.

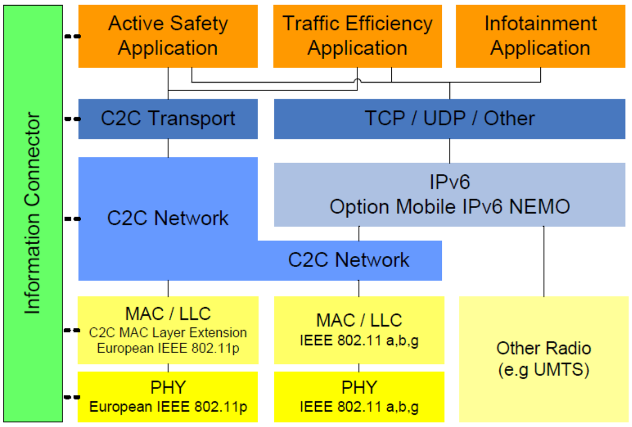


Fig. 1. Protocol architecture defined by the Car-to-Car Communication Consortium[1]

Multimedia data, specially video, if feasible, is very useful for entertainment, and it also will help to enhance navigation safety. For example, video on demand services could be very interesting during long travels in highways. Another example of video streaming services in vehicular networks are videos clips of nearby accidents or dangerous situations. These videos can provide drivers warning advertisements with precise information. This will allow them to make a more informed decision (whether to proceed or turn back) based on

ISBN: 978-1-61208-161-8

personal priorities and/or on vehicle capabilities.

While a huge number of video-related applications are expected to be deployed in a VANET, in this article we focus on video services where network mobility is involved, e.g., video on demand services. These video services will be deployed in environments where there exists a network infrastructure. Thus, a video server is placed in the infrastructure domain and vehicular nodes access to this server during a travel. In that case, vehicular nodes need global mobility to be reached from the Internet. In vehicular networks, packets may be corrupted and lost due to channel errors and collisions. These type of packet losses tend to be random and locally diverse and thus can be countered efficiently with a local recovery strategy. However, in a scenario where a video on demand service is offered, the main packet loss cause is the great amount of handoffs due to network mobility during the whole communication.

There exists several studies in the literature related with video streaming services in VANETs, such as [5], [6]. However, these studies are focused in video streaming applications where the communications take place among peers, i.e., inter-vehicular communications, and the analysis of how video streaming services are affected by ad-hoc routing protocols or medium access control protocols in vehicular networks. There also exists an article [7], that studies network mobility performance (e.g., packet loss rate and delay) in similar scenarios. This paper can be considered an extension of this work analyzing a specific application - video on demand services in vehicular networks - in the same context

The novelty of this paper is the analysis of video on demand services in a highway infrastructure scenario using real video applications in emulated vehicular networks and how network mobility protocols limit the quality of a video streamed. Firstly, we present a study for the potential deployment of video on demand services in vehicular networks where a Mobile IP solution is used. Then, we compare the results with a vehicular network where Fast Handovers for Mobile IP (FMIP) are used for seamless communications during network mobility handoffs [8]. Moreover, we analyze the quality obtained in the movie clip streamed and we measure the video degradation during communication blackouts.

The reminder of this article is organized as follows. In Section II, the tools used for the simulations are described. The video streaming performance evaluation is presented in Section III. The reference scenario is presented in Section III-A, and the simulations results are analyzed in Section III-B. Section IV concludes the paper.

II. Vehicular Network Emulation

Academia and industry use simulation tools to debug and test the reliability and QoS of several applications. This makes simulation a very important step towards the deployment of wireless communication networks. A simulation is only useful if the simulation results match as closely as possible with the testbed results. However, despite all the technological achievements and cutting edge research occurring in the field of mobile wireless networks, there are growing concerns regarding the reliability of results generated by wireless network simulators.

Emulation means the ability to introduce the simulator into a live network using a soft real time scheduler which tries to tie the event execution within the simulator with the real time. Emulation permits to test real time applications in a simulated network. The emulation is divided in three modules based on its functionality. Moreover, emulation provides a more realistic approach. In this sense, in this article we have developed an application to emulate video streaming over VANET. Figure 2 shows the modules of the emulation platform: application virtualization, network emulation and traffic mobility simulation.

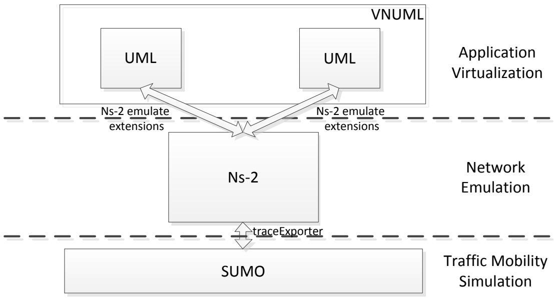


Fig. 2. Vehicular network emulation

The emulation modules are detailed in the paragraphs below:

A. Application Virtualization

The emulated network consist of a set of User Mode Linux (UML) [9] virtual machines running in a host machine. The UML virtual machines virtualize the network nodes. In our simulations, a UML machine represents the video server in the wired domain, and another UML machine represents a vehicular node in the wireless domain. The applications run inside the virtual machines and do not notice that they communicate through an emulated network, so the applications are executed as in a real-system. The network is emulated as a vehicular network transparently to the tested applications. UML virtual machines are managed using VNUML software [10].

B. Network Emulation

To emulate the network is used the widely known network simulator ns-2 [11] using the emulation feature, providing the ability to introduce the simulator into a live network and emulate a network that provides real applications in real time. This simulator is actively used for wired and wireless network simulations. We have introduced some to ns-2 in order to enhance its capabilities. In this sense, ns-2 Emulation Extensions [12] are used to enable ns-2 to emulate wireless networks using UML virtual machines. These extensions implement an interface between virtual machines and ns-2 using TAP devices [13]. They also improve the emulation of wireless networks in ns-2, enhancing the scheduler of the network simulator for the correct emulation of wireless networks. Another extension added to ns-2 is NO Ad-Hoc Routing Agent (NOAH) [14]. This extension emulates the behavior of a mobile node without using adhoc routing, so the mobile nodes only connect with the base stations. Finally, to provide FMIP support an extension developed by Robert Hsieh [15] is also added to ns-2.

C. Traffic Mobility

The last part of the emulation platform is the traffic mobility module. This module is responsible for creating the node movements such as a vehicle following the different itineraries defined by the road maps and the different configurable parameters, e.g., max speed limits, road lanes, crossroads, speed and acceleration of the cars, etc.

The traffic mobility module is mainly formed by SUMO [16]. The emulated network uses this traffic mobility simulator to provide node mobility traces to ns-2. This mobility traces provide to the network simulator/emulator information about the nodes positions and the speed of their movements necessary to calculate the network conditions. The mobility traces are obtained using the tool provided with SUMO software called traceExporter, which converts the dumps from SUMO to traces that can be used in ns-2.

III. Video Streaming Performance

Video streaming over vehicular networks can actually be applicable. The car engine can provide enough power for intensive data computation and communication. Vehicles can also be provided by large *On-board* storage. Thus the node in vehicular networks is powerful enough to forward continuous video data to other vehicles or roadside receivers. Furthermore, the IEEE 802.11g standard can support up to 54Mbps transmission rate, or the vehicular specific IEEE 802.11p [2] standard support up to 27Mbps. It is reasonable to expect a 1Mbps data rate between high speed driving vehicles within a highway using ad-hoc communications [17]. Therefore, using the transmission data rate required by compressed video, there is enough bandwidth to support video streaming for vehicles. However, the scenario analyzed in this paper involves communications between vehicles and the infrastructure. A vehicle, using a video player, is connected to a central video streaming server placed in the Internet. In this case, the handoffs between different subnets access points limit the expected bandwidth.

A. Reference Scenario

The test scenario designed for this purpose is an infrastructure scenario where a set of base stations are deployed over a highway in an overlapped manner. Therefore there are no coverage blackouts in the road. All the base stations are connected to a central router and this is also connected to a video streaming server. The base stations belong to different subnets, so every handoff in the scenario is a layer 3 handoff. The video streaming server and a vehicular node with a video player installed are emulated by UML virtual machines. The Mobile IP Home Agent is also placed in the video streaming server to simplify the scenario (see Fig. 3).

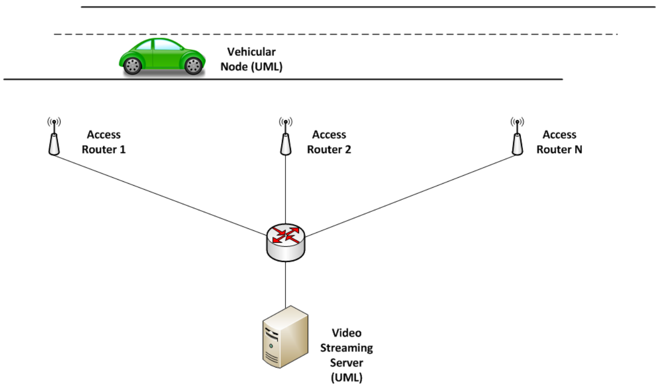


Fig. 3. Reference scenario

The routing between the video server and the car node is performed always as a single hop between the vehicular nodes and the base stations. Therefore no ad hoc routing is used in this testbed. The goal of the simulations is to carry out a study of a video streaming service over a highway with a lot of handoffs between base stations and analyze how video streaming services perform in a vehicular network using network mobility solutions. In Table I the parameters used in the simulation are detailed.

Live555 [18] is used to test the multimedia applications in the testbed. Using these libraries a video streaming server is configured in one side of the communication and a VLC media player [19] or a MPlayer [20], with live555 libraries to get real time features, in the vehicular node.

Parameter Name	Value
Wired links	Bandwidth: 100Mb Propagation delay: 5ms
Propagation model	Nakagami
Wireless access	IEEE 802.11p
Distance between APs	300m
Ad-hoc routing protocol	NOAH
Video characteristics	352x288 MPEG-2 CBR 500Kbps

TABLE I
SIMULATION PARAMETERS

B. Simulation Results

To check the deployability of a video on demand service over a highway, a set of simulations are performed. The main problem that can limit the deployment of a video service is the packet loss that occurs during the handoffs due to network mobility. The vehicle's high speeds in the roads and the amount of handoffs must be analyzed to deploy a video service.

1) Packet Loss: To analyze packet loss, a Constant Bit Rate (CBR) UDP traffic, without any Forward Error Correction

(FEC) or Automatic Repeat reQuest (ARQ) method, is sent from the server to the vehicular node, simulating a CBR class video streaming. To simulate this CBR stream and calculate packet loss, the Iperf tool [21] is used during 300 seconds per each bitrate and vehicle speed. Next graphs show the packet loss rate obtained using Mobile IP. The first graph, Figure 4, shows the packet loss rate of four different vehicular nodes speeds and its evolution when the CBR data rate is increased. The second graph, Figure 5, shows the packet loss rate using four different data rates, and the evolution when the vehicle speed is increased. It can be seen that packet loss rate increase as vehicle speed increase. In some cases the packet loss rate decreases for higher speeds. This is due the packet loss rate depends on the number of handoffs and it can be decreased increasing the vehicle speed. It also must be considered that the network reconfiguration time in Mobile IP during handoffs is an opportunistic value, in contrast with FMIP handoffs.

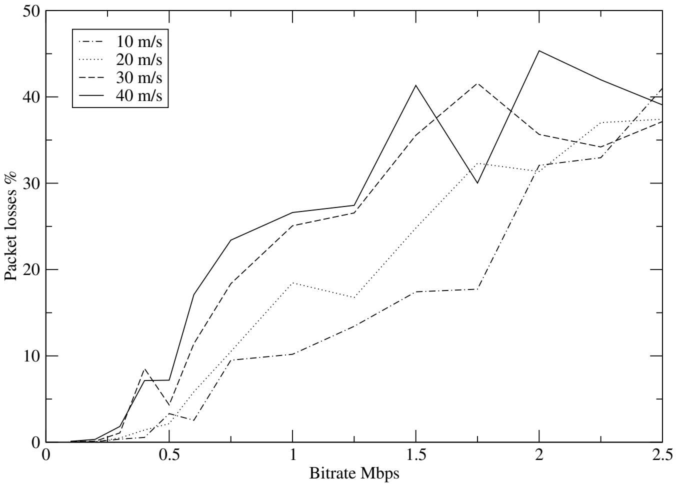


Fig. 4. Packet loss rate per bitrates using Mobile IP

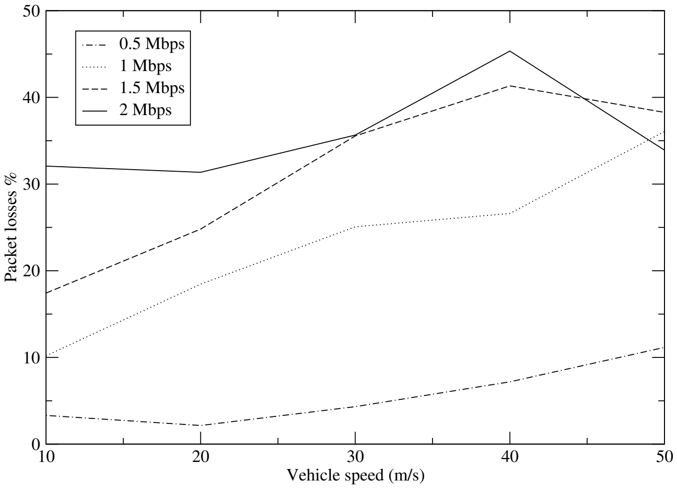


Fig. 5. Packet loss rate per speeds using Mobile IP

From Figures 4 and 5 it can be concluded that packet losses using Mobile IP solution are too high to deploy a video on demand service in a vehicular network. It is possible to appreciate that only for very low bitrates packet loss is acceptable. Therefore, Mobile IP is useless for a quality video streaming. Moreover, the handoff delay that follows the original Mobile IP can be up to seconds. For this reason, a protocol to get a seamless communication to an appropriate video reproduction is needed.

FMIP can reduce the handoff delay by either introducing L2 triggers to anticipate the handoff in advance or managing most of the handoff operations inside a local domain. It can be seen that FMIP protocol can reduce the handoff delay to get between 0.18 and 0.4 seconds in the 99.3% of the cases [7]. Minimizing the delay handoff, the FMIP standard reduces the amount of packet loss during the L3 handoff. In spite of FMIP's objective, it could not always guarantee the successful fast handoff if the moving speed of mobile node is very high. Since the L3 handoff of Mobile IP is controlled by the mobile node on a connectionless network, several messages should be exchanged among nodes to control handoff process, and handoff process of FMIP tightly depends on L2 triggering. These two features can increase the possibility of failure because the trigger does not consider the state of mobile node's L3 and delivers triggers only based on variable wireless signal state. So, although in FMIP the packets are buffered and supplied to the MN after the handoffs to avoid packet loss, these failures produce some packet loss that affect to the video streaming services. Figures 6 and 7 analyze the packet loss using the FMIP protocol.

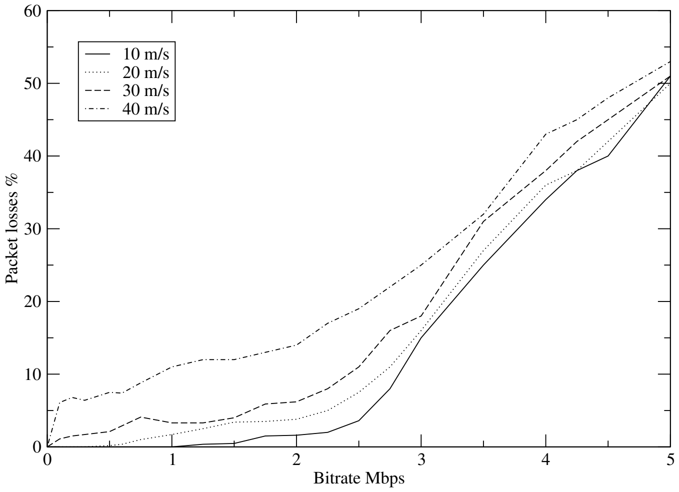


Fig. 6. Packet loss rate per bitrates using FMIP

Figure 6 suggests that using FMIP techniques video streaming can be feasible for 10, 20 and 30 m/s vehicle speeds. However, at 40 m/s some problem will occur while reproducing a video stream without any special technique due to the high packet loss rate. For 30 m/s, a video bitrate greater than 500 Kbps can produce some troubles. However, with this bitrate it is possible to reproduce a video with an interesting quality. For the last two speeds, 20 and 10 m/s, the problems arise at 1 and 2 Mbps. This means that for urban mobility a video

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

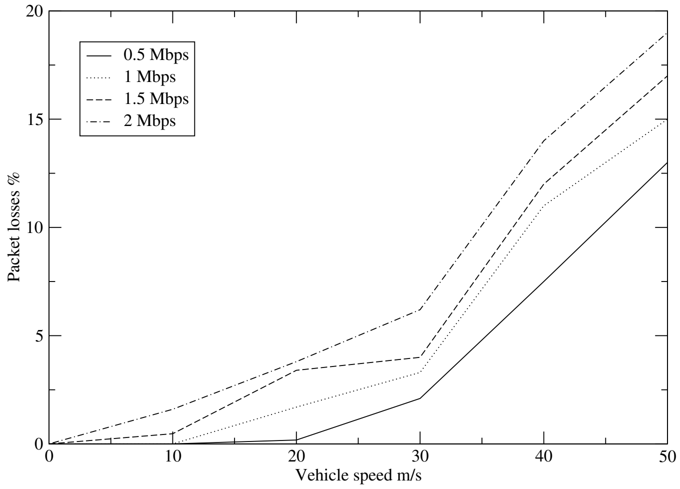


Fig. 7. Packet loss rate per speeds using FMIP

service can be deployed with a packet loss rate that can support an enough video rate to assure a high video quality and, for highway mobility also can be supported a video service, but with a lower video bitrate and a poorer video quality.

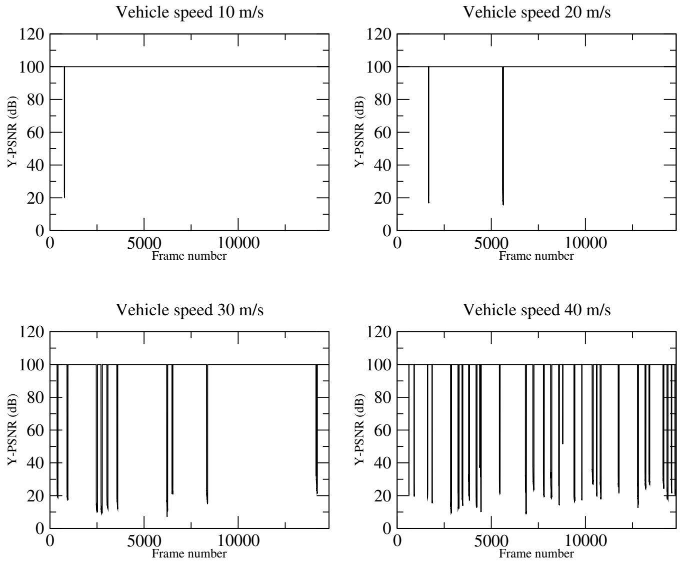


Fig. 8. PSNR of the video received compared with the transmitted video, for different vehicle speeds

2) Video quality: The objective of this test is to investigate how the quality of a video clip streamed in a vehicular network is affected by the handoffs occurred during the communication, measuring the quality of the video received compared with the original video using the Peak Signal-to-Noise Ratio (PSNR). The video used in the simulations is a 352x288 MPEG-2 video coded at 500 Kbps. To recover the gaps that are lost during the communication, an error-resilient decoder, based on an enhanced version of MPEG-2 decoder [22], is used. When this decoder is not able to recover the lost frame, the previous frame is represented. In Figure 8 the PSNR for different speeds is represented. When the video received is the same as the original video, the PSNR is represented as 100 dB. In this figure, the video quality degradation due to packet loss caused by the handoffs can be observed. For 40 m/s, a lot of gaps occur during the video reproduction. Therefore, this speed could be unfeasible to play a video during a travel using these scenario parameters. For slower speeds, as can be 10, 20 or 30 m/s, it can be observed that the quality degradation during the video reproduction is reduced drastically, so it could be feasible to play a video during a travel over a highway going at these speeds.

IV. Conclusions and further work

In this article, a set of simulations of live video streaming over vehicular networks have been presented. This set of experiments analyze the way handoffs limit the overall quality of a video streamed during a travel over a highway. The packet loss rate grows with the video bitrate and the vehicle speed increments, decreasing the video quality perceived by the client that is estimated with the PSNR of the decoded video. Although fast handoffs techniques to minimize handoffs blackouts are used, the packet losses limit the deployment of video streaming services in vehicular networks. For this reason it can be convenient in further research to analyze video streaming using reliable protocols to avoid packet loss during the communication. Further research will extend the video streaming analysis over vehicular networks to TCP transport protocol analyzing different TCP flavors behaviors.

Moreover, other network mobility techniques to prevent video streaming blackouts will be studied in further plans. This mobility proposal will present transport layer mobility instead of network layer mobility, including multi-path and multi-homing features and optimizing the communication data rate during the handoffs and preventing from network disconnections.

References

[1] "C2C-CC: CAR 2 CAR Communication Consortium," http://www.car-to-car.org/.

[2] S. Eichler, "Performance evaluation of the IEEE 802.11 p WAVE communication standard," in *2007 IEEE 66th Vehicular Technology Conference, 2007. VTC-2007 Fall*, 2007, pp. 2199–2203.

[3] C. Perkins, "Mobile IP," *International Journal of Communication Systems*, vol. 11, no. 1, pp. 3–20, 1998.

[4] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3963.txt

[5] F. Xie, K. Hua, W. Wang, and Y. Ho, "Performance study of live video streaming over highway vehicular ad hoc networks," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. IEEE, 2007, pp. 2121–2125.

[6] M. Bonuccelli, G. Giunta, F. Lonetti, and F. Martelli, "Real-time video transmission in vehicular networks," in *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007, pp. 115–120.

[7] Jiang Xie and Howitt, I. and Shibeika, I., "IEEE 802.11-Based Mobile IP Fast Handoff Latency Analysis," in *Communications, 2007. ICC '07. IEEE International Conference on*, vol. , no. , june 2007, pp. 6055 – 6060.

[8] R. Koodli, "Mobile IPv6 Fast Handovers," RFC 5568 (Proposed Standard), Internet Engineering Task Force, Jul. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5568.txt

[9] J. Dike *et al.*, "The user-mode linux kernel homepage," *URL: http://user-mode-linux.sourceforge.net*.

ISBN: 978-1-61208-161-8

[10] F. Galán, D. Fernándeza, J. Ruiz, O. Walid, and T. de Miguel, "A virtualization tool in computer network laboratories," in *Proceedings of 5th International Conference on Information Technology Based Higher Education and Training (ITHET'04)*, vol. 17, Istanbul, May 2004, pp. 27–34.

[11] "Network Simulator, version 2," http://www.isi.edu/nsnam/ns.

[12] D. Mahrenholz and S. Ivanov, "Real-Time Network Emulation with ns-2," in *In Proc. of DS-RT*, 2004, pp. 29–36.

[13] "VTun - Virtual Tunnels over TCP/IP networks," http://vtun.sourceforge.net.

[14] "NO Ad-Hoc Routing Agent (NOAH)," http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/.

[15] "Source code for ns-extension on HMIPv6 with Fast-Handover," http://mobqos.ee.unsw.edu.au/~robert/nsinstall.php.

[16] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner, "SUMO (Simulation of Urban MObility); An open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling (MESM2002)*. sn, 2002, pp. 183–187.

[17] J. Singh, N. Bambos, B. Srinivasan, and D. Clawin, "Wireless LAN performance under varied stress conditions in vehicular traffic scenarios," in *2002 IEEE 56th Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall*, vol. 2, 2002.

[18] R. Finlayson, "Live555 streaming media," 2009.

[19] "VideoLan - VLC media player - Open Source Multimedia Framework and Player," http://www.videolan.org/vlc/.

[20] "MPlayer - The Movie Player," http://www.mplayerhq.hu.

[21] "Iperf," http://sourceforge.net/projects/iperf/.

[22] "MPEG-2 Encoder / Decoder, Version 1.2, July 19, 1996," http://www.mpeg.org/MSSG/.

Vehicle Speed Estimation using Wireless Sensor Network

Muhammad Saqib, Sultan Daud Khan, Saleh Mohammad Basalamah
Department of Computer Engineering
Umm ul Qura University, Mecca, Saudi Arabia
{msskhan, sgkhan, smbasalamah}@uqu.edu.sa

***Abstract*—A real time locating system automatically tracks and localizes people and objects. We propose a model for the speed monitoring of vehicles using wireless sensor network-based on real-time localization. Our model is based on symmetric double sided two way ranging algorithm which has the ability to zeros out the effect of clock drifts between transmitter and receiver. In our model two anchor nodes are used as road side units at fixed locations and heights and moving vehicle is equipped with on board unit called a tag. Collected data from tag is used to calculate the speed of the vehicle at the base station. Several experiments are done to evaluate the performance of the proposed model at different sampling intervals of time by moving the vehicle at different speeds. Due to the noisy nature of collected data, discrete Kalman filter is used for better estimation of the speed of the vehicle. We have compared the true values of speed with measured values and estimated values. Experimental results show that the estimated values became close to the true values by applying Kalman filter.**

Keywords-SDS-TWR; Kalman filter; Traffic monitoring.

I. INTRODUCTION

Due to advancements in electronics sensor nodes become smaller and cheaper, while advancements in communication technology made the sensor node to communicate in a better way; for that several efficient protocols and algorithms have been developed [1]. These tiny sensor nodes are working together to make a wireless sensor network. Wireless sensor networks can be used efficiently in several applications like home automation, industrial control, military and health. Real-time localization is one such important application of wireless sensor network to locate objects.

In monitoring and surveillance applications, if the position of the reporting node is not known, the information we obtained is not useful information. Several nodes are deployed with known locations information; such nodes are called anchor nodes. Other nodes, which do not know their position information, are called blind nodes. Anchor nodes are used to locate blind nodes by using different localization algorithms. Ranging is the process to determine the distance between an anchor node and blind node. The distance is fed to localization algorithm to find the resultant location. Traffic surveillance is an important system which monitors the speed and traffic density. This information is useful for law enforcement agencies, to have a check on an over speeding vehicles, and also useful for transportation agencies to make transportation more intelligent by avoiding rush conditions.

In our paper, a system model is proposed for traffic monitoring using wireless sensor network. Traditionally, there are several techniques for traffic monitoring like inductive loops detectors and speed cameras. Inductive loops are installed in pairs in a travel lane for direct speed measurement. However, most of the time single loop detectors are used which cannot measure speed directly; estimation is used for such measurements [2]. Single loop speed estimation can be broadly divided into two types: (1) g-factor approach, and (2) stochastic filtering approach [2]. In g-filtering measured occupancy and volume is used to know the speed of vehicle while in stochastic approach Kalman filter is used for estimation of speed. Video systems used for traffic monitoring generally involves two tasks (1) Estimation of road geometry (2) Vehicle detection. Measurement taken by such system is then matched with the assumed road or vehicle model to determine vehicle speed and position. Such system must have low processing time, cost and high reliability which are difficult to get because of the high computationally expensive process like segmentation. Wu *et al*. [5] present a new algorithm that takes advantage of the digital image processing and camera optics to automatically and accurately detect vehicle speed in real-time. In [6], the algorithm is based on WSNs, which work according to the characteristics of the actual traffic stream, an on-road speed estimation model and algorithm based on wireless magnetic sensor networks was researched. In this model, 3 sensor nodes were working together to estimate the speed of passing vehicle. Totally different approach is followed in [8], which uses acoustic wave pattern to estimate vehicle speed. The acoustic wave pattern is determined using the vehicle's speed, the Doppler shift factor, the sensor's distance to the vehicle's closest-point-of-approach, and three envelope shape (ES) components, which approximate the shape variations of the received signal's power envelope.

In this paper, we propose a different approach from vision based systems. Our proposed system is based on a wireless sensor network, in which a vehicle equipped with on-board unit tag is continuously monitored by fixed access points. Our system is simple and inexpensive system, which can be further enhanced to include many other functionalities.

The rest of the paper is organized as follows; Section II describes the symmetric double sided two way ranging algorithm. Section III gives the proposed system model and discrete kalman filter design. Experimental results and discussion are given in Section IV. Section V discusses conclusion and future work.

II. Symmetric Double Sided Two Way Ranging (SDS-TWR)

Ranging is the most fundamental technology used in Real Time Locating Systems (RTLS). Nanotron Technologies developed SDS-TWR, which finds the distance between two nodes by measuring RToF symmetrically from both sides [4]. In some ranging systems, fine quality of clock generating crystal is required for measuring distance between two nodes. However, a cheaper way is that transmitter node calculates the round trip time to receiver node. Similarly, the receiver node calculates round trip time to transmitter node, and the resultant is average of two round trip times. The symmetric and double sided nature of transmissions, zero out the effect of clock drifts between transmitter and receiver even using the cheap oscillator on both sides.

In SDS-TWR, measurement of time starts at node A by sending a ranging request to node B, as shown in Fig.1. Node B starts its time measurement by receiving the packet from node A, and stops when it sends a reply to node A. Node A calculates the round trip time from the accumulated time in the received packet from node B. The difference between the measured time by node A minus the measured time by node B, equals to the twice of the time of signal propagation. Similarly, in the second measurement at node B, which sends ranging request to node A and starts its time measurement. Node A starts its time measurement, when it receives the packet, and stops, when it replies with a packet to Node B. Propagation time t_p is given by equation (1).

$$t_p = \frac{t_{roundA} - t_{replyA} + t_{roundB} - t_{replyB}}{4} \qquad (1)$$

III. Proposed System Model and Discrete Kalman Filter Design

In our proposed system for traffic monitoring, two anchor nodes with known locations and at known heights H, are used. Anchor nodes are called road-side units (Access Points), which are denoted by anchor1 (AP1) and anchor2 (AP2).

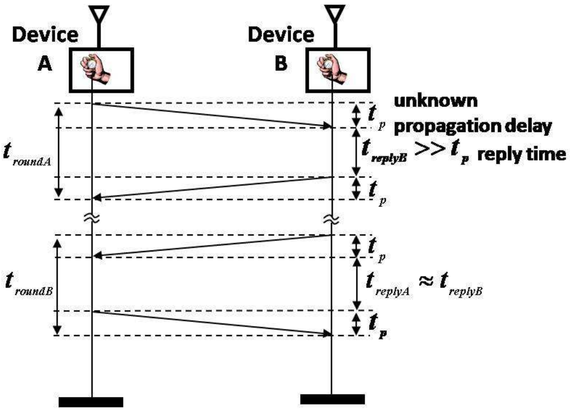


Figure 1. Symmetric Double Sided Two Way Ranging

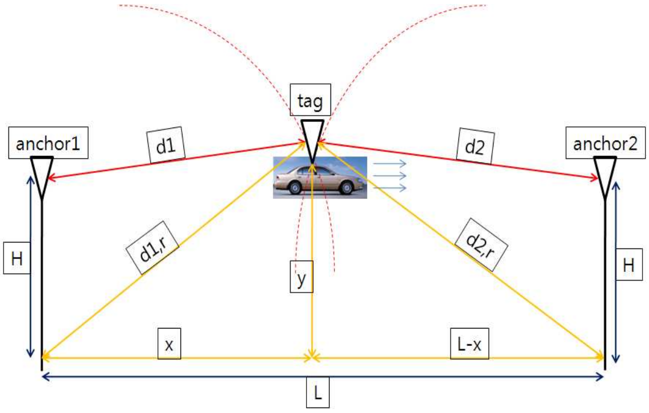


Figure 2. Proposed Model

Moving vehicle is equipped with on-board unit called tag, which is communicating with both anchors, and also with base station. The base station is used to collect data from the tag, and calculates its position and velocity. Total distance between two anchor nodes is measured manually, denoted by L. At any moment, the distance traveled by moving vehicle is denoted by L-x, where x is the traveled distance. Ranging distance between AP1 and tag is denoted by d1 and between AP2 and tag is given by d2, where $\mathrm{d1_r}$ and $\mathrm{d2_r}$ are real distances and can be calculated as,

$$\mathrm{d1_r} = \sqrt{\mathrm{d}_1^2 - \mathrm{H}^2} \qquad (2)$$

$$\mathrm{d2_r} = \sqrt{\mathrm{d}_2^2 - \mathrm{H}^2} \qquad (3)$$

$$\mathrm{x} = \frac{\mathrm{d1_r^2 - d_{2r}^2 + L^2}}{\mathrm{2L}} \qquad (4)$$

$$\mathrm{y} = \sqrt{\mathrm{d1_r^2} - \mathrm{x}^2} \qquad (5)$$

First time reading: $(x_1,y_1),t_1$

Second time reading: $(x_2,y_2),t_2$

$$v = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \Big/ (t_2 - t_1)$$

where x and y are the coordinates of the moving vehicle and v is the resultant speed of the vehicle. Our proposed system is a linear dynamic system. Kalman filter is used to get a better estimation out of noisy measured data [7].

The Kalman filtering process can be divided into the following steps.

Step 1: State Equation of the linear dynamics system is given below.

$$X_{k+1=}AX_k + \Gamma u_k + W_k \tag{6}$$

$$\binom{x}{v}_{k+1} = \begin{pmatrix}1 & T\\ 0 & 1\end{pmatrix}\binom{x}{v}_k + \begin{pmatrix}T^2/2\\ T\end{pmatrix}u_k + W_k \tag{7}$$

Where Position is given by

$$x_{k+1=}\, x_k + T.v_k + \frac{T^2}{2}u_k + w_{k.x} \tag{8}$$

Velocity is given by

$$v_{k+1=}\, v_k + T.u_k + w_{k\,.v} \tag{9}$$

$w_k \sim N\,(0, Q_k)$, where w_k is process noise with zero mean and Q_k covariance (in our experiment, it is acceleration uncertainty). A and Γ are transition matrices. T is sampling interval, and U_k is constant input to the system.

Step 2: Measurement equation

$$Y_k = CX_k + Z_k \tag{10}$$

$Z_k \sim N\,(0, R_k)$, where Z_k is the measurement noise with zero mean and R_k covariance. Measurement noise mainly occurs due to instrumentation errors, while C represents the transition matrix.

Step 3: A priori error covariance and Kalman gain are given in "(11)" and "(12)" respectively. Where $\hat{P}_k$is the initial estimation covariance.

$$\overline{P_{k+1}} = A\hat{P}_k A^T + \Gamma Q_k \Gamma^T \tag{11}$$

$$K_{k+1} = \overline{P_{k+1}}C^T(C\overline{P_{k+1}}C^T + R_{k+1})^{-1} \tag{12}$$

Step 4: A posteriori state estimate

$$\binom{\hat{x}}{\hat{v}}_{k+1} = \binom{\bar{x}}{\bar{v}}_k + K_k\left[Y_{k+1} - C\binom{\bar{x}}{\bar{v}}_k\right] \tag{13}$$

Velocity and position after estimation are shown in "(13)".

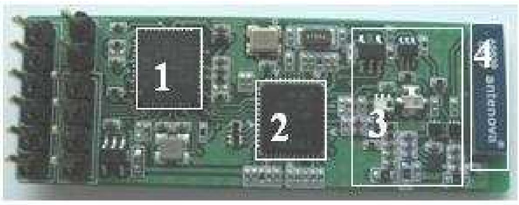


Figure 3. Redesigned module with added Power Amplifier (20mmx50mm)

IV. Experimental Results and Discussion

Performance of the proposed system model is evaluated using nanoLOC TRX Transceiver by Nanotron Technologies [4]. We have redesigned the hardware module by adding a power amplifier to it, as shown in Fig.3. The module contains ATMEGA 644 (1), NanoLOC (2), power amplifier (3), and chip antenna (4). NanoLOC TRX transceiver is used for base station, AP1, AP2, and tag operating in ISM 2.4 GHz frequency band. Proposed system model is evaluated by driving a car equipped with on-board unit (tag) at different speeds, e.g., 20,40,60,80,100,120 km/hr between two APs. During each experiment, the vehicle was moved at a constant velocity. Experimental results are corrupted by noises. There are two noises to cater for.

- Process noise: The velocity is perturbed by noise due to gusts of wind, potholes.
- Measurement noise: Measurement noise is mainly due to instrumentation errors.

Kalman filter exploits the predictable nature of the uncertainty or noise in the signal to optimize the estimation, based on the prediction of the model as well as updates from the measurements. Several parameters of the model are the following.

- Operating range of each device is (400m-700m).
- Height of rod at which AP1 and AP2 are installed is 9.5m.
- Total distance between AP1 and AP2 is denoted by L=356m.
- Several parameters required for kalman filter to work properly are process and measurement noises, which are obtained by carefully observing experimental and actual data.
- Process noise= 4m/sec^2, and measurement noise=5m.

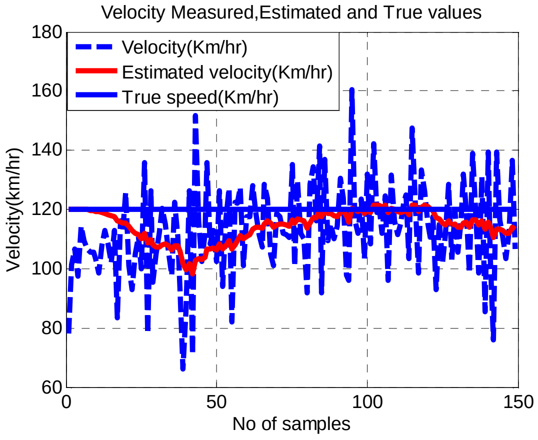


Figure 4. Speed of the vehicle at 120 Km/hr.

In Fig.4, the speed of the vehicle moving with constant speed of 120 km/hr is monitored. A solid blue line indicates the true speed of the vehicle, which is kept constant between two AP's. True speed of the vehicle is given by the vehicle speed meter. The experimental data obtained from the sensors are shown by the blue dotted line in the graphs, which are corrupted by noises. Deviation from the true speed can easily be seen in the graphs. It is difficult to maintain constant speed between two APs, which may also be the source of noise in experimental data. Noises are in the form of acceleration uncertainty. Kalman filter is used on experimental data for estimation, which is shown by the red line in Fig.4. Estimated results are closer to the true speed of the vehicle. Results show that tuning of kalman filter is required to obtain best results out of corrupted data.

Experimental data are taken at different sampling intervals of times, e.g., 90ms, 450ms. Performance is evaluated by comparing the measured data with true data and estimated data from a kalman filter.

In Fig.6, when the vehicle is moving with the lower speeds, e.g., 40 km/hr, then measurement and estimated results are closer to the true results, but at higher speeds, experimental data are more prone to noise and error.

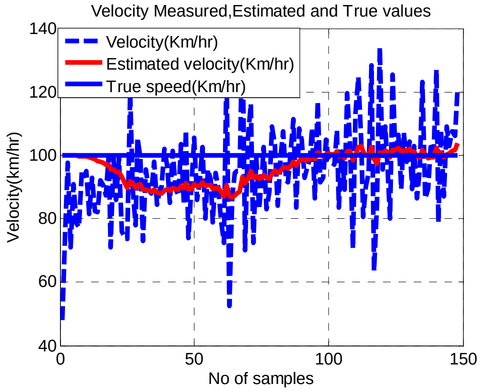


Figure 4. Speed of the Vehicle at 100 Km/hr.

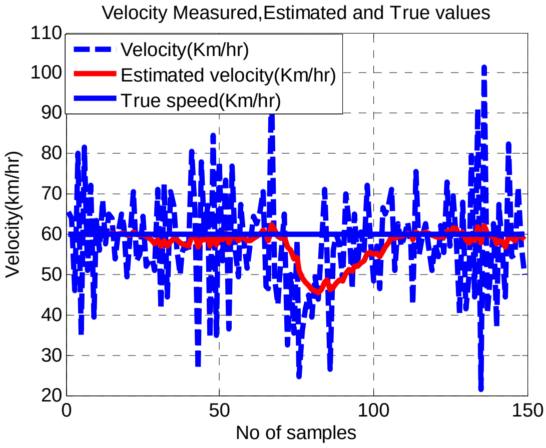


Figure 5. Speed of the Vehicle at 60 Km/hr.

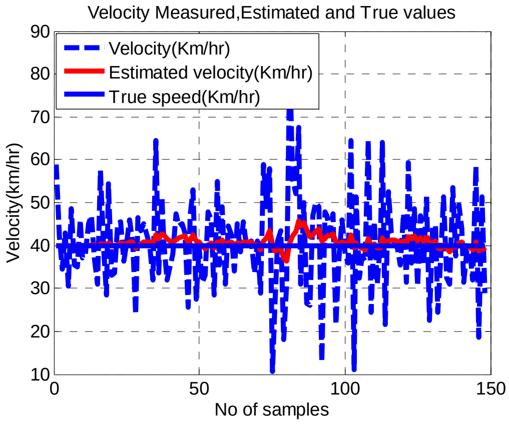


Figure 6. Speed of the Vehicle at 40 Km/hr.

V. CONCLUSION AND FUTURE WORK

In this paper, model for traffic monitoring system using a wireless sensor network has been proposed, and practically evaluated using Nanotron sensor boards. In the proposed model, whenever the moving vehicle with tag appears in between two anchor nodes, its position and velocity are determined and displayed on the base station. Kalman filter is used for better estimation. The proposed model has several advantages over existing systems; it is more robust and requires less computation than existing systems with expensive cameras. Model is evaluated for single vehicle, but it can be easily extended for many vehicles.

REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "Wireless Sensor Networks: A Survey", Computer Networks (Elsevier) Journal, vol. 38, no. 4, pp. 393-422, March 2002.

[2] Jianhua Guo, Jingxin Xia, Brian and L.Smith, “Kalman filter approach to speed estimation using single loop detector measurement under congested conditions” J. Transp. Engrg. Volume 135, Issue 12, pp. 927-934, December 2009.

[3] V. Kastrinaki, M. Zervakis, K. Kalaitzakis, “A survey of video processing techniques for traffic applications” Image and Vision Computing, Volume 21, Number 4, April 2003, pp. 359-381(23).

[4] http://www.nanotron.com/EN/pdf/WP_RTLS.pdf

[5] Jianping Wu, Zhaobin Liu, Jinxiang Li, Caidong Gu, Maoxin Si, “An Algorithm for Automatic Vehicle Speed Detection using Video Camera” Proceedings of 2009 4th International Conference on Computer Science & Education.

[6] Ding Nan, Tan Guozhen, Ma Honglian, Lin Mingwen, and Shang Yao, “Low-power Vehicle Speed Estimation Algorithm based on WSN” Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008.

[7] G. Welch and G. Bishop, “An Introduction to the Kalman filter” Technical Report TR 95-041, Computer Science, UNC Chapel Hill, 1995.

[8] Volkan Cevher, Rama Chellappa and James H. McClellan, “Vehicle Speed Estimation Using Acoustic Wave Patterns” IEEE Transactions on Signal Processing, vol. 57,Jan 2009.

Testbed architecture for generic, energy-aware evaluations and optimisations

Alexander Kipp, Jia Liu, Tao Jiang, Jochen Buchholz, Lutz Schubert
High Performance Computing Centre Stuttgart (HLRS)
Stuttgart, Germany
{kipp | liu | jiang | buchholz | schubert}@hlrs.de

Micha vor dem Berge, Wolfgang Christmann
Christmann Informationstechnik + Medien GmbH
Ilsede, Germany
{micha.vordemberge | wolfgang.christmann}
@christmann.info

***Abstract* - Worldwide data centers CO2 emissions are equivalent already to about the total airlines' CO2 emissions and are expected to overcome the 40% of Total Cost of Ownership of worldwide IT by 2012. Data centre electricity consumption accounts for almost 2% of the world production and their overall carbon emissions are greater than both Argentina and the Netherlands. Since computing demand and electricity prices rise, posing new environmental concerns, and due to limited resources, energy consumption of IT systems and data centre energy efficiency are expected to become a priority for the industry. In particular within the HPC domain the continuously raising energy consumption is seen as a major issue to be addressed. Therefore, new approaches are required facing this challenging issue. In this paper, we are going to present the setup of a testbed architecture and realisation in order to enable the evaluation of a broad range of differing (infrastructure) environments whilst keeping the maintenance efforts as little as possible. Therefore, the presented testbed architecture allows for the best possible inspection of the entire testbed environment for evaluation issues, in particular with respect to the evaluation and comparison of the energy efficiency of different approaches and environmental settings.**

Keywords - Testbed architecture; Evaluation; Energy-Efficiency; High Performance Computing; Monitoring; Management.

I. INTRODUCTION

This paper is framed within the GAMES (Green Active Management of Energy in IT Service centers) project [4], which is targeting to develop methodologies, models, and tools to reduce the environmental impact of such systems. The focus of this paper is on a new approach taking into consideration the classification of the energy efficiency of HPC applications, by allowing for an effective monitoring solution of complex IT infrastructures enabling to analyse the according energy consumption in a fine-granular way, whilst keeping the affect of this monitoring processing as little as possible for the entire infrastructure. In particular within complex infrastructures with a large amount of compute nodes to be monitored these systems often are supervised just in a sporadic way in order to provide as much of the available compute power to the running applications and services. For this reason, common systems do typically not allow determining potential energy wasting since the according overview about the energy consumption of specific compute nodes, in particular in combination with the according executed applications, is missing.

Within GAMES we already defined a set of so called "Layered Green Performance Indicators" [9] [6] to analyse the energy efficiency of applications and on actions that can be undertaken to save energy such as redundancy elimination from applications or better exploitation of middleware and processing infrastructures. Our set of Green Performance Indicators regards various system components (CPU, memory, I/O channels, and so on); once an energy leakage has been discovered through monitoring in one or more components, green actions allow one to (at least partially) remove or reduce this energy loss by reducing redundancies of data and processes, by adjusting the storage subsystem e.g., by running disks in slower mode, or by improving the rate of CPU usage, just to mention a few of several possibilities.

The GAMES approach framing this paper proposes guidelines for designing and managing applications along the perspectives of energy awareness. The approach focuses on the following two aspects, with the ultimate goal of developing a new systematic scientific discipline in the area of Green Computing:

a) the co-design of energy-aware information systems and their underlying services and IT service centre architectures in order to satisfy users requirements, performance, QoS, and context whilst addressing energy efficiency and controlling emissions. This is carried out through the definition of Green Metrics, enabling us to evaluate if and to what extent a given service and workload configuration will affect the IT resources footprint;

b) the run-time management of IT Centre energy efficiency, which will exploit the adaptive behaviour of the system at run time, both at the application and IT architecture levels, considering the interactions of these two aspects in an overall unifying vision.

In this paper, we are going to provide an overview of the GAMES testbed infrastructure allowing for the evaluation of complex IT infrastructures, in particular with respect to the according energy-efficiency of the hardware environment as well as the corresponding applications.

We are furthermore introducing a fine-granular monitoring architecture of the according infrastructure whilst keeping the payload on the corresponding systems as little as possible, as well as allowing for the easy adaptation of the entire testbed with differing images and configurations. In particular the latter aspect allows the easy comparison of differing testbed setups.

Copyright (c) IARIA, 2011.   ISBN: 978-1-61208-161-8

Therefore, we are going to provide an overview of the architecture of our flexible testbed environment in Section II. In Section III we are giving an overview about the technical details of our testbed environment, and, in particular, of our monitoring approach. Details about the extractable monitoring information are given in Section IV. Finally, conclusions are given in Section V.

II. TESTBED ARCHITECTURE

First of all, we present the architecture of the testbed, as being depicted in Figure 1. The testbed consists of a GAMES cluster system, internal network server, imager server, Nagios server, Network-attached storage (NAS) and a frontend server. All these parts are connected via a Gigabit interconnect network, which is going to get enhanced by an Infiniband interconnect with the delivery of the next generation of the RECS (Resource Efficient Computing System) [3], which has been applied within our testbed environment providing the best suitable solution for monitoring complex environments. As the essential part of the entire testbed, the cluster system is supposed to collect the required information for the GAMES framework by monitoring the energy consumption, application status and other obligatory parameters, whilst executing submitted jobs and deployed services [7]. More importantly, the runtime controller of the GAMES framework is able to adjust the configuration of the testbed and leverage workloads among computing nodes according to the adaptation methodologies.

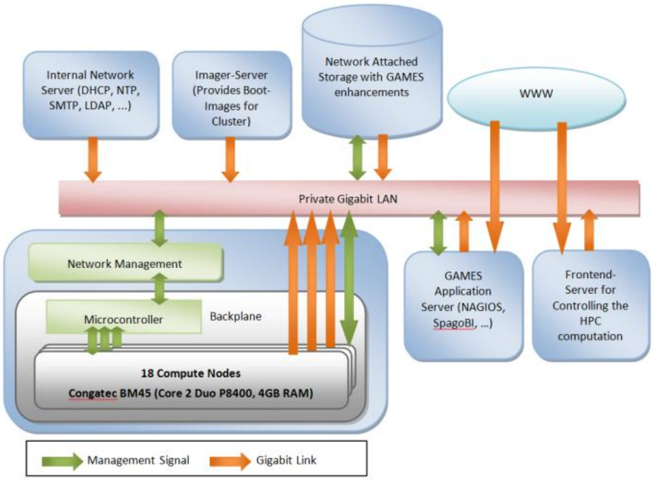

Internal Network Server (DHCP, NTP, SMTP, LDAP, ...)
Imager-Server (Provides Boot-Images for Cluster)
Network Attached Storage with GAMES enhancements
WWW
Private Gigabit LAN
Network Management
Microcontroller
Backplane
18 Compute Nodes
Congatec BM45 (Core 2 Duo P8400, 4GB RAM)
GAMES Application Server (NAGIOS, SpagoBI, ...)
Frontend-Server for Controlling the HPC computation
Management Signal
Gigabit Link

Figure 1. Testbed Architecture

A. *Conceptual Testbed Setup*

Before presenting the cluster system, let us introduce the other infrastructure machines as well shortly. For designing the cluster's infrastructure we followed some basic principles:

- Low maintenance effort needed running the whole system.
- Main goal is to offer high performance computing capacities.
- Separate basic services i.e. user management and DHCP or a batchsystem for job. So if any of these services may fail, the cluster itself is still operational.
- Isolation of the end users in a way that they may use any software they want but are not able to corrupt any vital services.
- Additionally it should be possible to simply add more nodes or service machines to create more scenarios besides the HPC ones

These principles lead us to some specific implementation aspects

- All compute nodes are operating without local discs. So it can be guaranteed that all nodes can use the same image and changes in this image are available to all compute nodes immediately.
- Frontends should also use the same image since it would be an obstacle if the user has to prepare his job on a node different from the target node.
- We use read-only images for all computing nodes, therefore no user can accidently change anything. Even a possible attacker may not be able to use most existing exploits since they have to change some files, which is not possible within the mentioned setup.
- All service infrastructure machines may have local disc and boot from them since these services (i.e., Nagios) should be available even if some other services (especially the image server) are not available.

The decision to abandon disks from the compute nodes and frontends causes additional effort bringing the overall system online but when the infrastructure is once completely operational it is much easier to switch between different scenarios and even exchange the operation systems between jobs. This proceedings allows for the easy evaluation and comparison of differing scenarios and settings.

By using pre-created images for the clients, only the links to the images have to be changed for a specific node and after the reboot this node boots from the new image. It has to be noted that these images are executed natively on the according compute nodes, so there is no lost of performance due to an intermediary virtualisation layer whilst allowing for easy changes between different configurations and system setups. So it is also possible to provide different images in different nodes at the same time and – if virtualisation is used– even multiple images on one host are possible. For complex scenarios the cluster can be changed within minutes requiring just a very small overhead in the system wide configuration since the different scenarios are configured within the images and not in the global system.

At the same time it is possible to add writeable space through mount points and store the according logging information on the infrastructure server, so that in case of errors the log information is not lost. Our design also allows some minor node-specific adjustment during start-up since each node executes a start-up script referencing some node-specific files if they exist. This feature is mandatory, especially for the frontend server allowing for a smooth and, according to the system configuration of the compute cluster, consistent testbed configuration.

Taking all this into account we have designed a testbed architecture allowing for the required flexibility whilst considering the monitoring of the entire testbed environment in a fine-granular way. In particular the latter aspect is quite important for evaluation testbeds in order to afford for detailed evaluations of the according executed tests. Within this architecture, the *Frontend Server* offering worldwide access for end users. On these hosts the users are enabled to prepare their jobs, compile their sources, upload data and submit their jobs in the first place. Later on there might be a cloud frontend where they can deploy their services or compute jobs. The frontend server is used by all logged in users to work in shared mode so that the processing speed depends on the activity of other users at the same time. This implies that the frontend server is not designed to be used for intense computing, but only for preparing and testing.

The computation jobs are executed on the *Compute Nodes* where the user's jobs are scheduled by a batch systems so that only one user uses a node at a time – at least for the HPC scenario. In a cloud scenario this might be different. Details about the compute nodes and especially the monitoring are described in the following.

Since the frontend and compute nodes are diskless we use *a Network attached Storage (NAS)* to provide storage capabilities in several ways. Therefore, all diskless nodes mount their root directory from the NAS server. Additionally the home directory of all users is provided to all nodes by the NAS server via the NFS protocol. In practise it might happen that many users do not care about their data after they finished their jobs and so the storage capacity will soon be reached. In order to avoid this, we restrict the space in the home directories and introduce workspaces which have a limited lifetime. This allows us to ensure that the users have enough storage for their jobs and the space will be freed again after they finished saving their data, so the users are taken into responsibility to decide which data is sensible to be stored for later usage, and which data is just of temporary nature (e.g., intermediary results of a simulation process), which can be erased after a specific period of time.

The above mentioned systems are the main parts of the system visible to the user. However, in order to handle and manage the entire testbed environment, a dedicated *Infrastructure Server* has been set up controlling the entire cluster. This server provides DHCP and acts as a log host for the diskless clients, account management, a batchsystem and scheduler for the HPC scenario, a SMTP Gateway so that any node – especially GAMES services and monitoring, can send eMail-notifications. So this server is the critical part in running the cluster and therefore no other services should be deployed on it to ensure a smooth operation of the infrastructure.

For additional services, test environments, etc. we installed an additional server – the *GAMES Server* - where currently Nagios as well as the according GAMES infrastructure components are installed in order to collect the desired monitoring information, compute the according metrics including power consumption and temperature to built the database on which decisions can be made on how the nodes should be used to reduce the energy footprint, as well as the GAMES runtime environment allowing for adaptive actions in order to optimise the entire energy consumption of the testbed cluster [8].

Finally, within our testbed architecture we considered an *Imager server* which is used to update / create images without interference to the other systems. Therefore this imager server mounts a copy of the old image in read-write mode and after the new image is ready all nodes can mount the new image in read-only mode. This imager can be also one of the compute nodes but to provide a constant number of compute nodes we do this on a dedicated machine.

This concept allows us to create images in parallel to the working environment, distribute these images to the compute nodes – not necessarily the same for all nodes – and control these nodes from a central point, namely the infrastructure server. In the same way we can add additional frontends of services on the same host to be accessed from the outside.

B. Enhanced Monitoring Infrastructure

In order to evaluate the described testbed in operation, a fine-grained monitoring solution has been developed allowing a detailed inspection of the entire testbed. In order not to influence the evaluation results by bothering the testbed environment with the according monitoring actions, a new monitoring infrastructure was evaluated within a research project funded by the German federal ministry for economy and technology [11], in which the University of Paderborn and Christmann designed the RECS [3]. The concept of this monitoring approach is to reduce network load, avoid the dependency of polling every single compute node at operation system layer and build up a basis on which new monitoring- and controlling-concepts can be developed. For these needs there is an especially designed central backplane for the RECS with an integrated master-slave system of microcontrollers.

The status of each compute node within the GAMES test cluster is connected to an additional independent microcontroller in order to manage the measured data. The main advantage of the RECS system is to avoid the potential overheads caused by measuring and transferring data, which would consume lots of computing capabilities, in particular in a large-scale environment this approach can play a significant role. On the other hand, the microcontrollers also consume additional energy. Comparing with the potential saved energy, it is expected that the additional energy consumption could be neglected. This microcontroller-based monitoring architecture is accessible to the user by a dedicated network port and has to be read out only once to get all information about the installed computing nodes. If a user monitors e.g., 10 metrics on all 18 nodes, he would have to perform 180 pulls which can now be reduced to only one, because the master does a pre-aggregation and processing of the monitoring data. This example shows the immense capabilities of a dedicated monitoring architecture. Further technical details are provided within the following section.

In order to allow for the monitoring of the energy consumption of the infrastructure servers, an external power meter has been installed in the environment as well, following a similar type of microcontroller based architecture is used for the power meter. The power meter is a CLM5-IP-P from Christ Elektronik and can monitor five single power lines, two thermal sensors, as well as some additional digital signals. All these information can be read out with the help of a self developed Nagios plugin with only one pull. Further technical details are described in the GAMES Deliverable D5.4 in section 2 [5].

III. Technical Testbed Realisation

The cluster server consists of 18 single PCs. The mainboards of the current RECS are COM Express based Congatec BM45 modules with an Intel P8400 CPU (2x 2.26 GHz, 1066 MHz FSB) and 4GB DDR3 Dual Channel RAM, each mounted on a baseboard which makes it possible to use almost every available COM Express mainboard. With the availability of the new Sandy Bridge architecture from Intel for COM Express the computing nodes could be scaled up to Quadcore i7 with max. 16 GB RAM. Each baseboard is connected to a central backplane which connects the Gigabit Network Interfaces to the front panel of the server. The mainboards have a loose interconnection regarding the central monitoring infrastructure but the main connection is the Gigabit Network Interface with a 24 Port Gigabit Switch.

TABLE I. List of Measured Data

Input Data	Data Source	Unit
*Status of the Mainboard	RECS MicroController	On/Off
Network Link Present	RECS MicroController	Yes/No
Network Speed	RECS MicroController	10/100/1000 MBit/s
Network Link Active	RECS MicroController	Yes/No
Fan Rotational Speed	Mainboard-Sensor	Rotations Per Second (rpm)
*Temperature of the Mainboard	CMFB4000104JNT, RECS MicroController	°C
Temperature of the CPU	Mainboard/CPU-Sensor	°C
*Current used by the Mainboard	ACS715ELCTR-20A-T, RECS MicroController	Ampere
*Voltage of the Power Supply Unit	ATMEGA169P-16AU ADU, RECS MicroController	Volt
*Power consumption of the Mainboard	RECS MicroController	Watt
Potential on the Mainboard	Mainboard-Sensors	Volt

Remark: Sensor Data that theoretically can be quantified by the Cluster System; due to missing mainboard support only marked (*) entries can be quantified with the actual system

All components within the Cluster Server share a common Power Supply Unit providing 12V with an efficiency of more than 92%. The several potentials needed for the mainboard chipset, CPUs and other components are provided by both the baseboards and the mainboard potential transformers.

The novel monitoring technique of the Cluster System introduced in the previous section is realized by a dedicated master-slave microcontroller architecture which collects data from connected sensors and reads out the information every mainboard provides via SMBus and I^2C. Each baseboard is equipped with a thermal and current sensor. At the current state not all sensor data that could theoretically be captured is available due to limited support of the mainboards. A list of theoretically measureable data is given in TABLE I. . All sensor data are read out by one microcontroller per baseboard which acts as a slave and thus waits to be pulled by the master microcontroller. The master microcontroller, and thus the monitoring- and controlling-architecture, are accessible to the user by a dedicated network port and additionally by a LCD display at the front of the server enclosure.

Additionally to the monitoring approach, the described infrastructure can be used to control every single compute node. Right now it is possible to virtually display the power- and reset-button of each mainboard. This enables the GAMES framework to control more energy saving states of the hardware than being possible with common systems, because the framework can wake up sleeping compute nodes and turn on completely switched off nodes. Of course it is even possible to have a mixed setup of energy consumption where some nodes are under full load, others are completely switched off and some nodes are waiting in a low-energy state for computing tasks. The following energy states are theoretically possible for every single compute node:

- **On, Maximum Performance**
  CPU frequency 2.27 GHz, no CPU throttling, Linux scheduler at maximum power state
- **On, Low Performance**
  CPU frequency 800 MHz, CPU throttling, Linux scheduler at energy saving state
- **Sleeping/Hibernate**
  CPU off, RAM in low power state
  Due to missing Linux support the actual system cannot be put into this state
  but future systems should be able to reach this state
- **Off**
  Completely switched off, turn on via the microcontroller

This flexibility in adjusting the testbed environment during runtime allows for the evaluation of differing setups quite easily, in particular with respect to the enhanced monitoring environment allowing for a detailed analysis of the according effects on the entire behaviour, in particular with respect to the according energy consumption. Within GAMES we are going to evaluate in how far the described new monitoring- and controlling approach can be seen as an enabler to monitor and control IT systems at a very fine granularity whilst keeping the payload for the computing environment as little as possible. Furthermore it has to be evaluated what mixture of energy states provides the best balance between maximum performance and energy efficiency. Refer to D5.4 [5] for further information and the corresponding metrics.

IV. MONITORING OF THE TESTBED INFRASTRUCTURE

At the time of writing this paper, the Nagios-based GAMES Monitoring infrastructure is configured to provide an initial information base for the GAMES framework prototype, in particular for the data mining and knowledge management activities. In particular, this monitoring information is the backbone for any kind of analysis and optimisation of the environment in order to achieve specific goals. As shown in [1] and [2] this monitored information, used to compute the according Green Performance Indicators [6], allow for an extensive analysis and optimisation of the according hardware and software settings. The testbed is monitored by a central Nagios instance and the monitor data are stored in a data base. The initial monitoring data set on energy consumption is composed of following metrics:

- *Mainboard Temperature:* Temperature of each single board (sensor is placed below the mainboard).
- *Central Voltage:* input voltage which is the same for all compute nodes installed on the base board.
- *Power:* power consumption in Watt of each compute node.
- *Status of the Mainboard:* on or off
- *CPU Usage:* the CPU occupation of a compute node including %user, %nice, %system, %iowait %idle values.
- *Memory Usage*: Memory occupation of a compute node.
- *Process info*: resource usage of the simulation process including its CPU Usage, Memory Usage and CPU time.

In particular, the first four values are measured directly from the master microcontroller through one single pull by using one single NAGIOS plugin. The output of the collected information of the according plugin is structured as follows:

```
{OK, all 18 boards available|nr_boards=18
|boards_status=1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1
boards_temp=31;27;28;28;26;27;27;26;28;33;32;33;31;3
1;30;30;30;30
central_voltage=11.78
boards_power=23;24;24;21;21;19;21;19;18;11;20;14;16;
13;16;17;16;26}
```

The last three values are measured in a traditional way by using three different plugins. These metrics are collected from the RECS nodes and stored into a historical data base during the execution of certain simulation jobs. It has to be noted that the data being monitored is a first draft for the GAMES prototype, other metrics like Storage Usage, I/O per second/Watt, Application Performance (FLOPs/KWh) which are relevant for the evaluation, in particular for HPC environments, are planned and will also be monitored once the corresponding hardware and software modules are available. However, the flexible structure of our testbed allows also for the (future) integration of not yet available sensing equipment. Furthermore the monitoring data set can be flexibly extended on demand. Due to the limited space of in paper, we show only the monitoring data of two of eighteen nodes being occupied by the simulation process. Beside the evaluation of simulation processes we are also considering cloud computing environments based on the OpenNebula framework [12] within our environment. The according evaluation results are going to be provided at [4] due to space limitations in this paper. In Figure 2 the monitored energy and resource consumption related metrics of the according computer nodes are depicted. The left y-axis shows the percentaged usage of the according CPU and Memory, whilst the right y-axis shows the absolute values for the according energy consumption in Watt and the temperature in °C.

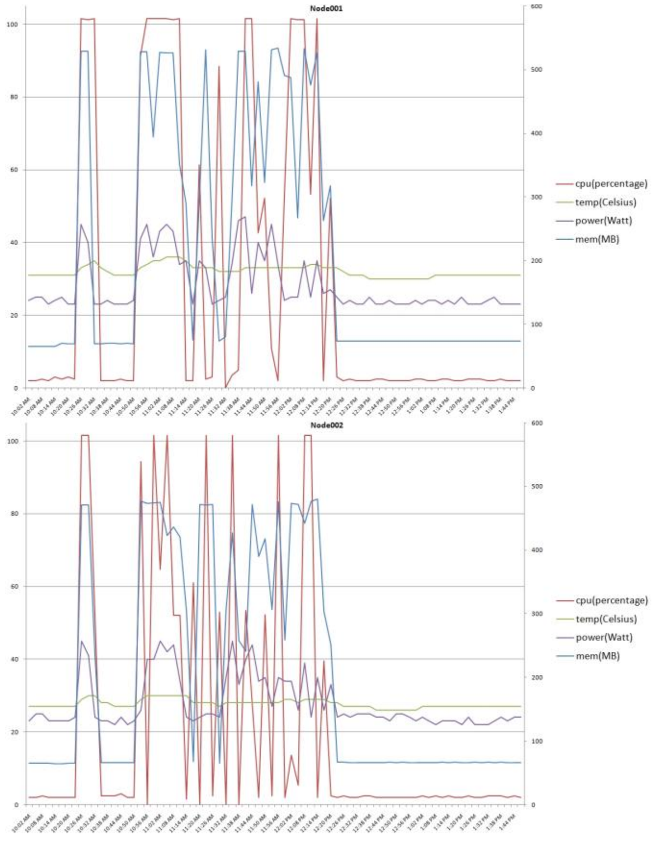


Figure 2. Monitoring Data on node 001 and 002

A detailed overview about the resource consumption is given in Figure 3. In this figure, the y-axis describes the percentaged usage of the according resources. This selected view on the according monitoring information allows for determining similarities between the resource consumption behaviours of different applications and services and varying platforms, which can be combined with the according monitoring data about the corresponding energy consumption. This proceeding allows for the determination of application profiles with respect to their energy consumption and IT resource utilisation. In particular, this aspect is reflected by the data mining solution of the GAMES framework, which will allow for an automated analysis and determination of these application profiles.

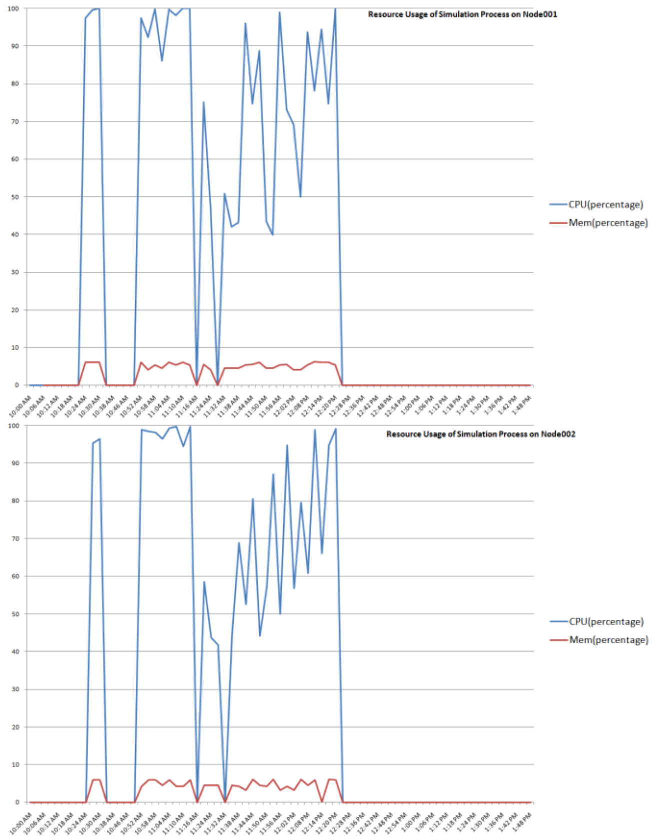


Figure 3. Resource Consumption Footprint of Simulation Process

V. Concluding Remarks

In this paper, we presented the architecture and setup of a testbed infrastructure, allowing for a flexible and dynamic evaluation environment. The presented architecture as well as the according setup faces the central requirements for an effective evaluation environment for differing setups and configurations, in particular with respect to the evaluation of the according energy efficiency, whilst keeping the required management effort as little as possible.

Acknowledgment

This work is partially supported by funding from the Information Society Technologies Program under the 7th Research Framework Program of the European Union (Call: FP7-ICT-2009-4 Grant agreement no.: 248514), within the research project GAMES (Green Active Management of Energy in IT Service centers). This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper. The authors want to thank all who contributed to this paper, especially all members of the consortiums.

References

[1] Chen D, Henis E, Cappiello C, et al. Usage Centric Green Performance Indicators. In: *Proceedings of the Green Metrics 2011 Workshop (in conjunction with ACM SIGMETRICS 2011)*. San Jose, California, USA; 2011:5.

[2] Cioara T, Anghel I, Salomie I, et al. Energy Aware Dynamic Resource Consolidation Algorithm for Virtualized Service Centers based on Reinforcement Learning. In: *Proceedings of the 10th IEEE International Symposium on Parallel and Distributed Computing (ISPDC 2011).*; 2011:8.

[3] Description for Resource Efficient Computing System (RECS) Available at: http://shared.christmann.info/download/project-recs.pdf [Accessed June 13, 2011]

[4] GAMES project website: http://www.green-datacenters.eu/ [Accessed June 13, 2011]

[5] GAMES Report D5.4 - GAMES Energy Efficiency Assessment Integrated Tool First Release; Available at www.green-datacenters.eu [Accessed June 13, 2011]

[6] Kipp A, Jiang T, Fugini M, and Salomie I. Layered Green Performance Indicators. *Future Generation Computer Systems*. 2011. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167739X11000860.

[7] Kipp A, Schubert L, Liu J, et al. Energy Consumption Optimisation in HPC Service Centres. In: *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*. Ajaccio, Corsica, France; 2011:1-16. Available at: http://www.ctresources.info/ccp/paper.html?id=6281. [Accessed June 13, 2011]

[8] Kipp A. GAMES - Green Active Management of Energy in IT Service centres. *inSiDE (Innovatives Supercomputing in Deutschland)*. 2010:40-43. Available at: http://inside.hlrs.de/pdfs/inSiDE_autumn2010.pdf. [Accessed June 13, 2011]

[9] Kipp A., Jiang T, and Fugini M. Green Metrics for energy-aware IT systems. In *Proceedings of the Fifth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS).In Press*. Seoul, Korea, 2011

[10] Liu, J. The Need for a Global CO2 Lifecycle Model in IT Service Centers. In: *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*. Ajaccio, Corsica, France; 2011:1-16. Available at: http://www.ctresources.info/ccp/paper.html?id=6282. [Accessed June 13, 2011]

[11] ZIM Erfolgsbeispiel Available at: http://www.zim-bmwi.de/erfolgsbeispiele/zim-koop-foerderbeispiele/zim-koop-025.pdf. [Accessed June 13, 2011]

[12] OpenNebula Cloud Computing Webpage. Available at http://opennebula.org/ [Accessed June 13, 2011]

Mptrace: Characterizing Physical Memory Usage for Chip Multiprocessors

Francesc Guim[†][*], Miguel Ferrer[*], Josep Jorba[*], Ivan Rodero[‡][*] and Manish Parashar[‡]
[†]*Intel Barcelona, Spain*
[*]*Open University of Catalunya, Spain*
[‡]*NSF Center for Autonomic Computing, Rutgers Univeristy, United States*
Email: {fguim, mferrer, jjorbae}@uoc.edu, {irodero, parashar}@cac.rutgers.edu

***Abstract*—The performance of high performance computing applications depends highly on how they are implemented. However, their runtime behavior is tightly coupled with the resources they are allocated such as in which cores the application threads run or in which memory devices their memory space is placed. Thus, depending on their characteristics, applications may exhibit more affinity to specific types of processors or perform better in a given architecture. In this paper, mptrace, a novel PIN-based tool aiming at profiling the applications physical memory usage is presented. Mptrace provides mechanisms to characterize the applications access to physical memory pages and thus to explore possible memory usage optimizations such as those aiming at improving performance-power trade offs. Two different implementations of mptrace are described and evaluated using NAS parallel benchmarks. The study of physical memory usage of NAS parallel benchmarks along with the discussion of a specific use case shows the large potential of mptrace.**

***Keywords*-PIN tool; memory profiling ; pagemap; NAS Parallel Benchmarks; High Performance Computing.**

I. Introduction

High Performance Computing (HPC) evolved over the past decades into increasingly complex and powerful systems. Reaching exaflops computing performance by the end of the decade require the development and deployment of complex and massive parallel processors with multiple cores (e.g., chip multiprocessors) and/or heterogeneous units [1] (e.g., the IBM/Sony Cell processor). The rapid increase in the number of cores has been accompanied by a proportional increase in the DRAM capacity and bandwidth, which presents many challenges such as performance-power trade offs and new programming challenges.

In order to achieve sustained performance and fully tap into the potential of these architectures, the step that maps computations to the different elements must be as automated as possible. In a coarse grain, applications can be classified as memory or processor bound. While the first type of applications is memory bandwidth greedy applications, the second one is mainly limited by either the processor parallelism level or by the amount of computational power that they require. In both cases, mapping computations to appropriate elements (e.g., physical memory) is an important task for two main reasons: (1) ensuring application's performance is crucial from the user perspective, and (2) maximizing the system utilization may improve the system throughput.

When applications use only a subset of all the resources available they waste a substantial amount of power and prevent other applications from taking advantage of the resources, and the system can perform different actions such as allocating the applications in the resources that best match their requirements or reducing the amount of power provided by the resources (e.g., using dynamic voltage and frequency scaling). However, the implementation of such techniques requires profiling methods that are fundamental to understand the applications behavior. Different existing tools provide mechanisms to instrument and gather runtime information. Among them, PAPI [2] provides an interface for collecting low level performance metrics (e.g., number of L2 misses) from hardware performance counters. Other tools are such as Intel PIN [3] provide information related to the application performance. They are able to intercept the application execution flow and to provide information regarding the application performance. Both type tools can be used to characterize the application in terms of processor performance (instructions executed), cache and memory performance (L1 hit rate, L2 hit rate, L2 misses per kilo instructions, etc.) and network usage (link utilization, etc.). Some of them also provide information regarding the virtual addresses used by applications, however, none of these tools provide ways to characterize accurately the physical memory usage and thus how the different channels to physical memory are used.

In this paper we present mptrace, a PIN-based tool, which is able to provide the physical pages that are used by processes on run time. It can be also used to extrapolate other meaningful information such as the usage of the different channels to physical memory. This can help to design new architectures and techniques to optimize the memory usage, thereby improving important aspects such as performance-power trade offs. The main contributions of this paper are: (1) the design and implementation of the mptrace tool, which extracts the mapping between the virtual memory address thread space and the physical memory space, and (2) the study of physical memory usage of NAS Parallel Benchmarks (NPB), which shows the large potential of mptrace.

The rest of the paper is organized as follows: in Section

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

II the background and related work are discussed; in Section III the mptrace tool is described in detail; in Section IV the evaluation using the NPB is provided as a use case; finally, conclusions and future work are discussed in Section V.

II. Background and Related Work

Previous approaches tackled the characterization of applications mainly from a performance perspective. Existing tools such as PAPI [2], Vampir [4], Paraver [5], Intel vTune [6] and PPW [7] allow to instrument applications by gathering runtime information for both application and computing resources. Existing approaches characterized how applications perform on top of specific hardware. Tools like PAPI, Vampir or Paraver allow instrumenting applications and gathering hardware counters for their executions and extracting information about how they behave. These tools are especially interesting to detect regions of the application that can be improved or to detect system bottlenecks. Other tools do not require instrumenting the application. For example PIN-based tools [3] are able to run non-instrumented binaries and intercept all the stream of instructions prior their execution. These tools are able to gather information concerning the instruction that is about to be executed (i.e., instruction type, operands, etc.).

Over the last years processors have evolved to become very energy efficient supporting multiple operating modes and thus power management techniques have become subject of study. At a very coarse level, power management at server systems level has been based on monitoring load and shutting down unused clusters or transitioning unused nodes to low power modes [8]. Dynamically varying the voltage and frequency proportional to system load has also proved to be effective in reducing energy consumption [9][10]. Dynamic Voltage and Frequency Scaling (DVFS) provides power savings at the cost of increased execution time. Other approaches conducted the power management techniques at the processor level. For example, Cai et al. [11] propose a DVFS techniques based on the hardware thread runtime characterization. These approaches have been developed on top of tools that allowed them to dynamically gather information about the system and the applications. However, the data used to apply DVFS techniques is only from the processor, network and cluster.

The previous approaches tackled the energy consumption optimizations focused on the computing elements. However, memory devices have begun to significantly contribute to overall system energy consumption, and like processors, DRAM devices currently have several low power modes. Delaluz et al. presented software and hardware assisted methods for memory power management. They studied compiler-directed techniques [12], as well as OS-based approaches [13] to determine idle periods for transitioning devices to low-power modes. However, this is not going to be effective in multi-core systems. Cho et al. [14] studied assigning CPU frequencies for DVFS that are memory-aware because the focus of all prior work was on optimal assignment of frequencies to CPU ignoring memory.

Existing tools have already provided mechanisms to understand how applications use the main memory. Some of the previously discussed tools provide information about the hit rate that applications have in L2. This information can be combined with other metrics, for instance the cycles per instruction to estimate the bandwidth that the application requires to the memory (for instance using the misses per kilo instructions). Other trace-based tools can be used to get similar information. For example the PIN-based tool CMPSim [15][16]. It is a PIN [3][17] tool that intercepts memory operations that are fed to a chip multiprocessor cache simulator. The model implements a detailed cache hierarchy with DL1/IL1, UL2, UL3 and memory, and can be configured to model complex cache hierarchies (e.g., a SMP machine of 32 cores sharing the L2 and L3). However, all the previous tools cannot provide more detailed information about how the physical memory is used, such as the memory bandwidth requested to specific memory channels. To do this, these tools would need to provide information about which physical memory locations are mapped to the virtual regions for the application process.

III. Mptrace

The Intel PIN [3][17] project aims to provide dynamic instrumentation techniques to gather information about the instructions that applications execute. PIN API provides mechanisms to implement callbacks that are called once specific events occur on the execution of the target application (i.e., execution of memory operation). Thus, a PIN tool can be build on top of this API to collect a subset of all the available information. This tool can be executed with different applications and there is no biding with specific binaries. Thus no instrumentation is required to the target application of study.

As of today, many tools have been build on top of PIN such as CMPSim [15][16], which is a cache hierarchy simulator that intercepts memory accesses and simulates its accesses using a cache model. Other tools that profile the applications memory access can be found in the PIN software development kit. However, no PIN tool or similar instrumentation tool has been provided to profile the physical memory accesses that applications request. Mptrace is a PIN-based tool that allows intercepting the processes memory accesses and translating the virtual addresses to physical addresses.

Mptrace has two different mechanisms to translate the virtual addresses to physical addresses. The first one is based on the *pagemap* file system, which is a relatively recent mechanism in linux kernel. This file system provides information about the physical location for the given virtual address of a process. The second mechanism is based on a linux kernel module that translates the virtual addresses to physical addresses without requiring an operating system that supports the *pagemap* file system. Specifically, the kernel module uses *ioctl* system calls to obtain the address for a given page, and does the *walkpage* through the linux memory structures to do the translation. In the following subsections the two mechanisms are discussed in more detail along with the description of the information provided by mptrace.

A. The pagemap version

The first mechanism uses the *pagemap* file system to translate the virtual address to physical address. The *pagemap* file system was released in the kernel 2.6.25 and can be accessed through the */proc/pid/pagemap* filesystem. As it is described in the kernel source, this file allows a user space process to find to which physical frame each virtual page is mapped. It contains a 64-bit value for each virtual page, containing the following data (from *fs/proc/task_mmu.c*, above *pagemap_read*):

- Bits 0-54 page frame number (PFN) if present
- Bits 0-4 swap type if swapped
- Bits 5-54 swap offset if swapped
- Bits 55-60 page shift (page size = 1 “$\ll$” page shift)
- Bit 61 reserved for future use
- Bit 62 page swapped
- Bit 63 page present

Using the *pagemap* system, the mptrace PIN tool provides several functionalities to characterize how the applications access the physical memory pages. The format and information required is highly customizable, it provides information related to cache access (way and set), and memory accesses (physical page address). It also provides ways to reduce the amount of generated information, such as sampling and trace disabling when the application loads data, or the caches are warming up. The current implementation of mptrace provides mechanisms to characterize the memory accesses on the flight. Thus, this PIN tool can provide summarized information about how an application is using the main memory. For example, it provides page access histograms, or clusters of memory regions accessed during an interval of time.

B. The kernel module-based version

The second mechanism has been designed to allow operating systems that do not support the *pagemap* file system, and to improve the mptrace performance as is shown in Section IV. A new kernel module has been developed to carry out the translation of the virtual addresses to physical addresses. To do this it emulates *pagewalk* and process all the different structures provided by the linux memory management unit (MMU) to do the translation. The translation procedures provided by this module are mapped onto specific *ioctl* address. The mptrace kernel module translates a given virtual address to a physical address following the steps listed below.

- Mptrace contacts to the mptrace kernel module using the *ioctl* system call (*IOCTL_GG*) to get the translation for the virtual address @x.
- The kernel module performs the following actions to process the translation:
  - Given the process identifier provided by the user space it looks for the *mm_struct* which contains the information concerning the memory allocated to it.
  - Using the *pgd_offset* kernel function gets the page global directory for @x.
  - Using the *pmd_offset* kernel function gets the page middle directory for @x.
  - Finally using, the *pte_offset* kernel function gets the page table for the address @x. Using the kernel function *pte_page* gets the struct page associated to this virtual address.
  - In order to get the unsigned integer coding the physical address for the resultant struct page, the module uses the function *page_to_phys*.
  - If no error has occurred in the translation the physical address for @x. For example, in those cases were the physical page for the given virtual address is not present, the corresponding error will be returned to the user space.

C. Information and functionalities

As has been discussed in the previous sections, mptrace intercepts all the memory access that the application performs and generates trace files containing information of these accesses. The most representative output data provided by mptrace is described below.

- Current access with respect to the global execution flow: number of global instruction, number of thread instruction, number of memory access instruction, and timestamp.
- Type of memory access: type of access, the instruction pointer for the given instruction, number of operands, and size of the operation in bytes.
- Physical resources used by this operation. For each virtual address it provides: the physical address, cache line and set used by this operation in L1 and L2, and physical page.

Mptrace provides some functionalities that allow both reducing the amount of data generated and summarizing the application behavior such as the number of accesses to the different physical pages. In order to reduce the intrusiveness of mptrace, the structures that it uses have been implemented in a light way fashion (e.g., using lightweight data structures). Moreover, tests to evaluate the level of intrusiveness of mptrace have been conducted. The main

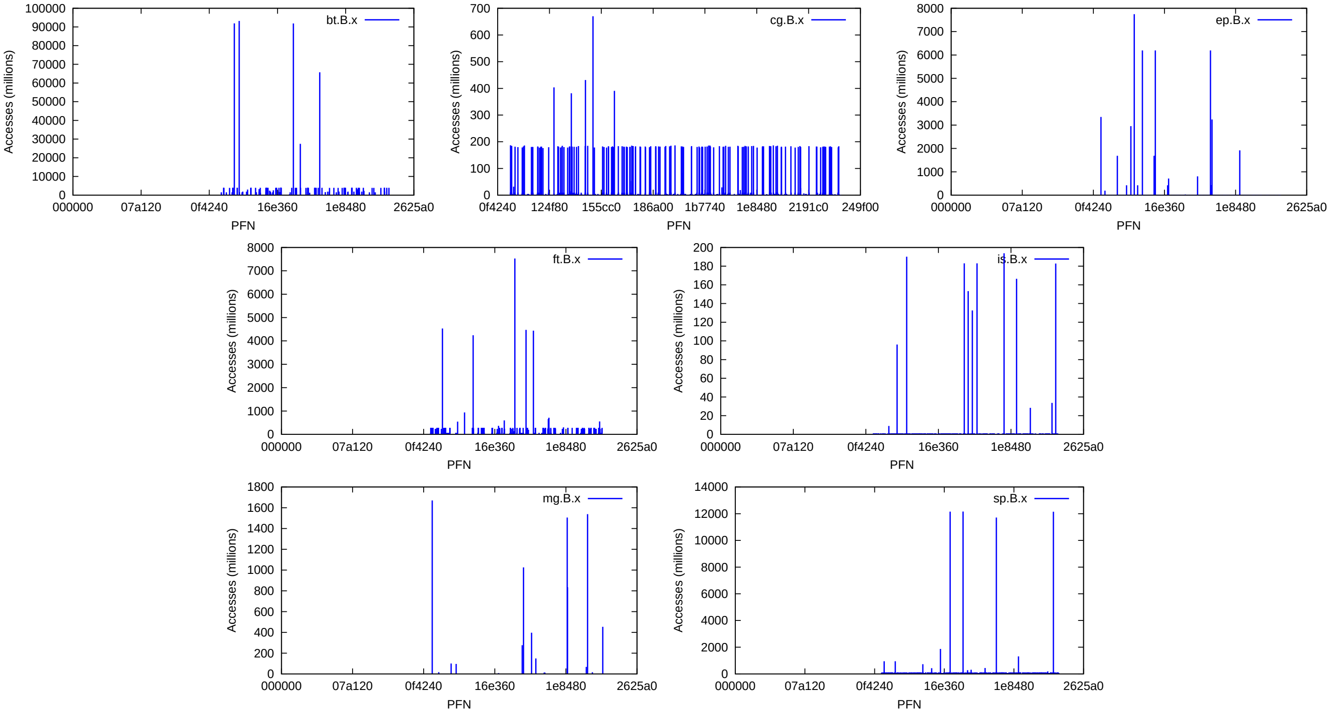


Figure 1: Access pattern of NAS benchmark: BT, CG, EP, FT, IS, MG and SP class B

goal has been to validate that the physical placement for the application virtual space is not modified by the fact that this PIN tool is running. Among other functionalities mptrace allows: sampling, specify in which intervals the memory request have to be processed, which threads have to be instrumented, counting specific events, and dump summarized information (i.e., the number of times that each physical page has been accessed).

IV. Evaluation

In this section, experimental results generated with mptrace are presented. The set of benchmarks that have been used are the NPB that are a set of benchmarks targeting performance evaluation of HPC systems. The goal of this study is to characterize how the different NPB kernels use the physical memory, and to understand how the working sets of these kernels are mapped to the physical pages by the operating system and hardware and how often these pages are accessed. A performance study of the two different implementations presented in this paper is also provided.

A. Methodology

The experiments were conducted with a server with an Intel(R) Core(TM)2 Quad CPU Q9450 processor and 8GB of memory running Linux kernel 2.6.34. The processor provides four hardware threads. NPB were run with the mptrace tool using the *pagemap* file system mechanism. The main parameters considered to conduct the experiments are listed below:

- Each application ran without co-allocation of other applications to avoid interferences with applications requesting memory to the operating system.
- Each application ran with the total amount of hardware threads that the processor provides in order to avoid context switching and other non-desired OS traps.
- Mptrace started tracing at the instruction count 1 million. In these experiments mptrace only accounted for the number of access to the physical pages and thus no other traces were generated (i.e., with the stream of reads and writes to the main memory).

B. Results

Figure 1 presents the number of accesses that each of the physical pages available in the memory device has been accessed by each of the NPB applications. The x-axes show the page number and the y-axes shows the millions of accesses that the application has accessed this page.

The plots show that the amount and distribution of memory access differ for the different NPB kernels. For instance, the MG kernel access few thousands of millions of memory instructions while the BT kernel memory accesses are more than 10 times larger. Since the MG and BT kernels run in 6 and 10 minutes, respectively, the amount of memory accesses per second is substantially different. However, this type of information can be gathered using other traditional tools (i.e., CMPSim). The interesting information that these plots provide is how separated the memory accesses for each of the NPB applications and the amount of accesses per physical page are from one another. In the case of the CG, the accesses are equally populated among all the different physical pages that are available to the threads. The rest of the benchmarks accesses are located in a relatively small number of physical pages. The MG, FT, EP, and BT benchmarks basically access to few tens of physical pages. However, the amount of accesses is very high for BT (up to 95,000 million access to the same page) with respect to the other two benchmarks (up to 1,800 million accesses to the same page). Therefore, three different type of patterns can be observed in this scenarios: CG does many accesses to many different physical pages; EP, LS and MG do small number of accesses to small subset of pages; and BT does large amount of access to a small subset of pages. Combining this information with time information and cache hierarchy information can lead to interesting characterization of how the memory subsystem is used. Furthermore, as it discussed in the following paragraphs it can derive to some optimizations in the memory system address decoder (i.e., how the virtual memory is placed in the physical memory) and how the memory device is configured (i.e., the amount of frequency that it has to run to deliver the required bandwidth).

As well as defining policies to address important problems such as reducing the memory contention when consolidating workloads, mptrace can be used, for example, to develop novel techniques such as predictive memory power management at run time. We propose using mptrace to extend the work on dynamic memory voltage scaling proposed by Deng et al. [18] considering the ability to select different frequencies for different memory channels as a case of study to show the large potential of mptrace. The process of mapping physical addresses to memory channels to main memory is proprietary to each memory control design. Mptrace can be used to obtain the physical addresses accessed by the applications and then process the data to obtain the channels access patterns. Figure 2 shows the memory access patterns for a large amount of channels (i.e., 64 channels) using two different algorithm for mapping memory physical addresses to channels: (1) *default*, where accesses are clustered to certain channels (e.g., clusters of 256MB), and (2) *interleaving*, where accesses are distributed across different channels. The figures illustrate how the algorithm for mapping physical memory addresses to channels can significantly affect the memory access pattern, and presumable the application behavior. They show that peak memory bandwidth is not always demanded by the application and there is unequal distribution of accesses across channels. This asymmetry and unequal distribution of traffic present opportunities to control the channels independently (i.e., scaling the dynamically the frequency).

In the previous sections two different mechanism to translate virtual addresses to physical address have been introduced: using the *pagemap* file system or using a kernel module. They do not only differ on the resources that they need but also they differ in their performance. As can be observed in Figure 3 the kernel-based implementation is substantially faster than the *pagemap*, especially for short and large runs. This figure presents the number of microseconds that mptrace needed to trace 100K, 1M and 10M of memory instructions for each of the NPB applications. In all of the cases the first implementation performs better than the second one. The difference is especially significant for 10k and 10M memory instructions. Hence the kernel implementation runs two times faster than the other implementation, on average, which is especially important for very large runs.

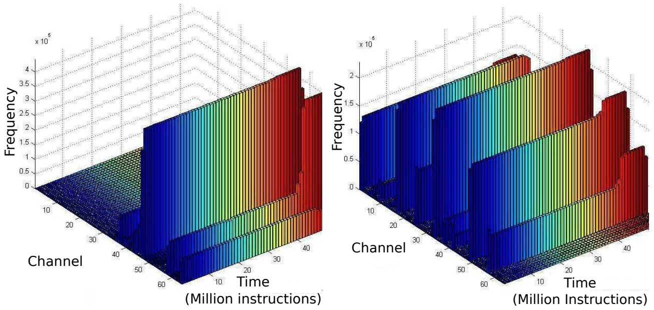


Figure 2: Channels access pattern with default (left) and interleaving (right) mapping policies

V. Conclusion and Future Work

In this paper, the mptrace PIN tool, which aims to profile and characterize the physical memory usage for HPC applications, has been presented. Two different implementations of mptrace are described and evaluated using NPB. Specifically, the physical memory usage is characterized by each of the NPB kernels. For each NPB kernel the number of accesses to each physical page is shown. Three different types of patterns are observed in this scenario: (1) CG accesses many times many different physical pages, (2) EP, LS and MG access fewer times a small subset of pages, and (3) BT accesses many times a small subset of pages.

The results show the large potential that mptrace has to study the applications physical memory usage. As of today, many of the tools provide mechanism to understand how the applications virtual space is used; however, information regarding the mapping of virtual addressed to physical memory allows us to understand how the memory devices are used (e.g., to understand the bandwidth required for each of the memory channels). This can lead to designing and optimizing novel architectures and software mechanisms along multiple dimensions such as performance, power and their trade offs.

Current and future research efforts include the development of: (1) a web-based framework to launch, process and generate memory characterization, (2) tools to automatically

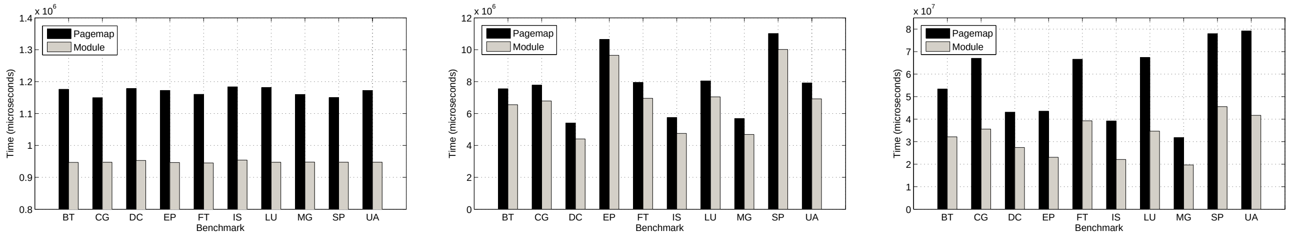


Figure 3: Execution tine of mptrace tracing 100K (left), 1M (middle), and 10M(right) instructions for NPB (OpenMP version)

characterize how the memory is used during the application execution, and (3) techniques to optimize the memory management based on the data provided by mptrace.

ACKNOWLEDGMENT

The authors would like to thank Jose Antonio Martinez, Miquel Perello and Karthik Elangovan for their contributions to this work. This work has been partially supported by the HAROSA Knowledge Community of the Internet Interdisciplinary Institute (http://dpcs.uoc.edu).

REFERENCES

[1] R. Kumar, D. M. Tullsen, N. P. Jouppi, and P. Ranganathan, “Heterogeneous Chip Multiprocessors,” *Computer*, vol. 38, pp. 32–38, 2005.

[2] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, “A Portable Programming Interface for Performance Evaluation on Modern Processors,” *Int. J. High Perform. Comput. Appl.*, vol. 14, pp. 189–204, 2000.

[3] C. Luk, R. Cohn, R. Muth, H. Patil, A. Klauser, G. Lowney, S. Wallace, V. J. Reddi, , and K. Hazelwood, “PIN: Building Customized Program Analysis Tools with Dynamic Instrumentation,” *ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2005.

[4] F. J. Gmbh, I. Bericht, W. E. Nagel, A. Arnold *et al.*, “VAMPIR: Visualization and Analysis of MPI Resources. http://www.pallas.de/pages/vampir.htm.”

[5] V. Pillet, J. Labarta, T. Cortes, and S. Girona, “PARAVER: A Tool to Visualize and Analyze Parallel Code,” In WoTUG-18, Tech. Rep., 1995.

[6] J. H. Wolf, “Programming Methods for the Pentium III Processor’s Streaming SIMD Extensions Using the VTune Performance tool,” 1999.

[7] H.-H. Su, M. Billingsley, and A. D. George, “Parallel Performance Wizard: A Performance System for the Analysis of Partitioned Global-Address-Space Applications,” *Int. J. High Perform. Comput. Appl.*, vol. 24, pp. 485–510, 2010.

[8] E. N. Elnozahy, M. Kistler, and R. Rajamony, “Energy-efficient Server Clusters,” in *2nd international conference on Power-aware computer systems*, 2003, pp. 179–197.

[9] N. Kappiah, V. W. Freeh, and D. K. Lowenthal, “Just in time dynamic voltage scaling: exploiting inter-node slack to save energy in MPI programs,” in *ACM/IEEE conference on Supercomputing (SC)*, 2005, p. 33.

[10] D. Zhu, R. Melhem, and B. R. Childers, “Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multiprocessor Real-Time Systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, July 2003.

[11] Q. Cai, J. González, R. Rakvic, G. Magklis, P. Chaparro, and A. González, “Meeting Points: Using Thread Criticality to Adapt Multicore Hardware to Parallel Regions,” in *International Conference on Parallel Architectures and Compilation Techniques*, 2008, pp. 240–249.

[12] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M. J. Irwin, “Hardware and Software Techniques for Controlling DRAM Power Modes,” *IEEE Trans. Comput.*, vol. 50, no. 11, pp. 1154–1173, 2001.

[13] V. Delaluz, M. Kandemir, and I. Kolcu, “Automatic data migration for reducing energy consumption in multi-bank memory systems,” in *39th Design Automation Conference (DAC’02)*, 2002, pp. 213–218.

[14] Y. Cho and N. Chang, “Memory-aware energy-optimal frequency assignment for dynamic supply voltage scaling,” in *International Symposium on Low Power Electronics and Design (ISLPED’04)*, 2004, pp. 387–392.

[15] A. Jaleel, R. S. Cohn, C. keung Luk, and B. Jacob, “CMPSim: A Pin-Based On-The-Fly Multi-Core Cache Simulator,” in *Fourth Annual Workshop on Modeling, Benchmarking and Simulation (MoBS)*, 2008.

[16] J. Moses, K. Aisopos, A. Jaleel, R. Iyer, R. Illikkal, D. Newell, and S. Makineni, “CMPSchedSim: Evaluating OS/CMP Interaction on Shared Cache Management,” in *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2009, pp. 113 –122.

[17] K. Hazelwood, G. Lueck, and R. Cohn, “Scalable Support for Multithreaded Applications on Dynamic Binary Instrumentation Systems,” in *2009 International Symposium on Memory Management (ISMM)*, Dublin, Ireland, June 2009, pp. 20–29.

[18] Q. Deng, D. Meisner, L. Ramos, T. F. Wenisch, and R. Bianchini, “MemScale: Active Low-power Modes for Main Memory,” in *6th International conference on Architectural support for programming languages and operating systems*, 2011, pp. 225–238.

Augmented Reality for Urban Simulation Visualization

Vincent Heuveline, Sebastian Ritterbusch, Staffan Ronnås
Engineering Mathematics and Computing Lab (EMCL)
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
{vincent.heuveline, sebastian.ritterbusch, staffan.ronnas}@kit.edu

***Abstract*—Visualizations of large simulations are not only computationally intensive but also difficult for the viewer to interpret, due to the huge amount of data to be processed. The case of urban wind flow simulations proves the benefits of mobile Augmented Reality visualizations, both in terms of selection of data relevant to the user and facilitated and comprehensible access to simulation results. This is achieved by a novel visualization method, presenting simulations based on current city model data correctly localized in real-world images.**

Keywords-Scientific Visualization; Augmented Reality; Numerical Simulation; Urban Airflow; Geographical Information Systems (GIS).

I. Introduction

The development of scientific computing including numerical simulation and interactive 3D visualization has today become an essential tool in many applications including industrial design, studies of the environment and meteorology, and medical engineering. The increasing performance of computers has played an important role for the applicability of numerical simulation but has also led to a data explosion. At present, the use of simulation software and the interpretation of visualization results usually require dedicated expertise. The large amount of data available leads to two problems for the end-user, which are discussed in this paper. On the one hand, handling and selection of the appropriate data requires a suitable user interface. On the other hand, the amount of perceptible information is limited, and thus visualizations of large data sets need very intuitive methods to be understandable.

Geographical Information Systems (GIS) are playing an increasing role for urban planning [1]. Their improved accuracy joined with the increasing performance of computing systems are making accurate large scale urban simulations feasible. In this paper we present the results of the joint work with the city council of Karlsruhe for simulations in an urban environment as an illustrative example setting, with focus on the advantages of mobile Augmented Reality visualization of large numerical simulations. The proposed visualization methods serves as a technology for solving problems of large scale data visualizations. Additionally, it also open the path to making results of numerical simulations accessible to decision makers and to the citizen at large, both from the technical and the comprehensional perspective. The general availability of smartphones and tables equipped with GPS, cameras and graphical capabilities fulfills the technical requirements on the client side for implementing the presented visualization methods. This allows for an intuitive exploration of large scale simulations. The ongoing standardization process of GIS for city modeling in the CityGML consortium [2] enable standardized simulation and visualization services for world-wide use based on the presented methodology in future.

The novel approach of providing scientific results on mobile devices was developed in the *Science to Go* project aiming to deliver numerical simulation on the spot.

In this paper, we present related papers and projects, and how they relate to the proposed concept. This is followed by the methodology of the visualization method, with details on the needed steps of preprocessing, simulation, augmented visualization, interaction and the server-client framework. The text ends with the conclusion and acknowledgements for partners and funding for the project.

II. Related Work

The Touring machine [3] was one of the first mobile solutions for augmented reality illustrating the potential of enhancing real life images in real-time for exploration of the urban environment. The approach was to display information overlays on the camera image, which is still popular in augmented reality applications of today [4], [5]. This concept is well suited to present textual or illustrative information, but is not directly suited for visualization of simulation results as presented in this paper. The availability of dedicated graphical processing units on mobile devices have led to augmented reality visualizations of pre-defined 3D objects [6] as they have been found beneficial in laboratory setups [7]. This is the basis for visualization of 3D structures representing the results of simulations. The use of augmented reality visualization for environmental data is presented in the HYDROSYS framework [8], providing a method to combine measurements and simulation data with geographic information. Similar to the work presented in this paper, this framework emphasizes the need for simulation information on-site. The conceptional need for combining simulation results with data from geographic information systems is also a driving force for the CityGML project [1], which has applications to natural disaster management. The augmented reality visualization of urban air flow phenomena in an indoor virtual reality laboratory setting based on physical mock-up building blocks is presented in [9].

III. Methodology

The integration of scientific visualizations into real world camera images is demanding from the perspectives of data preprocessing, mobile device positioning and the actual augmented reality visualization. The difficulties arise from the need to combine real-world and virtual geometries aligned with each other, and then to incorporate scientific visualizations of results from numerical simulations in the resulting image.

Simulating a phenomenon with a numerical method requires the computational domain to be determined. In our approach, the real world is represented by virtual city-models, which are converted into mesh data using sophisticated preprocessing techniques well known in the context of bio-medical simulations (see e.g., [10], [11]). Based on a mathematical model for airflow, a finite element CFD simulation is then set up and run using the HiFlow^3 simulation software [12]. Using accurate position and orientation of camera images based on sensor fusion techniques as discussed and illustrated in [4], a consistent Augmented Reality visualization of the simulation results can then be produced.

The proposed visualization methods for interaction with large numerical simulation on mobile devices are based on a client-server framework where specific demands have to be taken into consideration.

A. Preprocessing

The "3D-Stadtmodell Karlsruhe" [13] was started in 2002 as an improved database of geographic information to meet the demands of the urban administration. It consists of several data sets of varying purpose, coverage, accuracy and detail, starting with a terrain model without buildings, and including large brick models for the cityscape, up to a photo-realistic model, as seen in Figure 1. The city model is currently progressing towards an integration into a CityGML [1] based representation.

Since none of the models were created for use by numerical simulation software, extensive pre-processing steps are necessary. In general two or three models have to be combined to create a suitable computational domain, as seen in Figure 2. Special care was necessary to deal with model enhancements that had been made mainly for visual effects. For instance, there were closed window planes in garages facing the outside world on both sides with zero width, which are very significant for wind flow simulations around buildings. Although such irregularities could be avoided by imposing strict conditions on the city models, in general we cannot expect available city models to conform to these conditions, since they were originally created for visual planning. To avoid problems arising from these kinds of artifacts, an emphasis was put on the use of robust and performant region growing methods that are well known from medical applications such as for the realistic computational fluid dynamics simulations of the nose and lungs (see e.g.,[10], [11]).

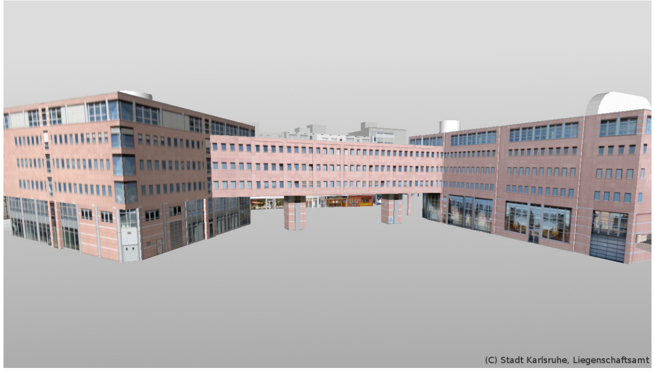


Figure 1. Photo-realistic building in the Karlsruhe 3D city model

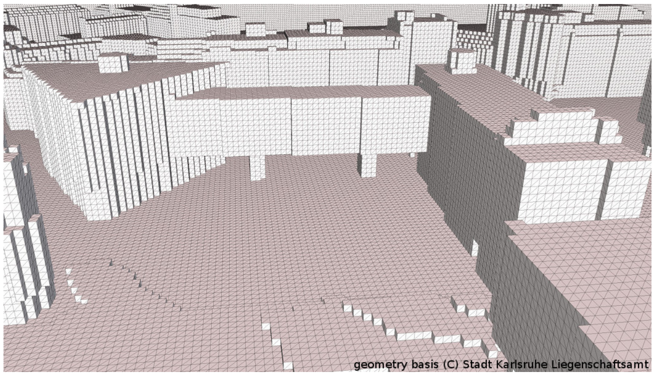


Figure 2. Computational geometry based on the Karlsruhe 3D City Model

Another challenge for enabling widespread use of simulation in urban environment is the non-availability of highly accurate city models. This condition can be weakened to the availability of high resolution models in the main areas of interest, since widely available low accuracy models are sufficient for the necessary peripherical simulation in the surrounding area. In spite of the varying detail of the models, the very accurate geographic alignment offers the opportunity for an automated data source selection and preprocessing workflow.

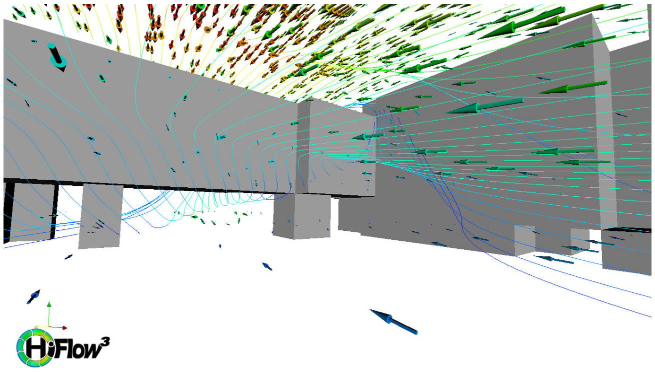


Figure 3. Numerical simulation results of urban wind flow

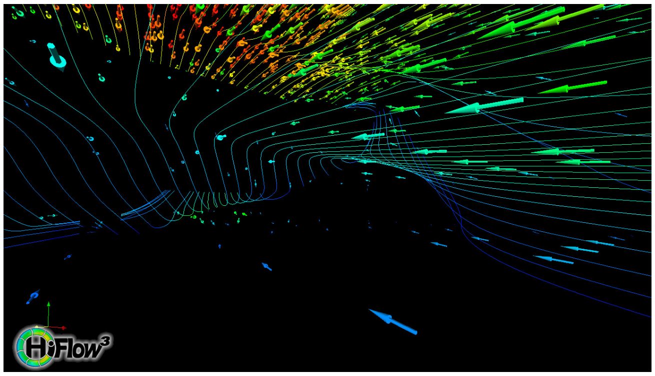


Figure 4. Masked numerical simulation visualization

B. Simulation

The instationary Navier-Stokes equations are solved in a sufficiently large computational domain surrounding the area of interest with suitable artificial boundary conditions for assumed wind flow conditions. At the walls of buildings the velocity is set to zero.

The model is formulated as an initial boundary value problem for the velocity $\vec{u}(\vec{x},t)$ and the pressure $p(\vec{x},t)$ in Equation 1.

$$\begin{aligned}
&\partial_t \vec{u} - \nu \Delta \vec{u} && \\
&+ (\vec{u} \cdot \nabla)\, \vec{u} + \nabla p = 0 && (\vec{x},t) \in \Omega \times (0,T) \ , \\
&\nabla \cdot \vec{u} = 0 && (\vec{x},t) \in \Omega \times (0,T) \ , \\
&\vec{u} = \vec{u}^{in} && (\vec{x},t) \in \Gamma_{\text{in}} \times (0,T) \ , \qquad (1)\\
&(-\mathcal{I}p + \nu \nabla \vec{u}) \cdot \vec{n} = 0 && (\vec{x},t) \in \Gamma_{\text{out}} \times (0,T) \ , \\
&\vec{u} = 0 && (\vec{x},t) \in \Gamma \times (0,T) \ , \\
&\vec{u}(\vec{x},0) = \vec{u}_0(\vec{x}) && \vec{x} \in \Omega \ .
\end{aligned}$$

The parameter ν in this model is the kinematic viscosity, which is assumed to be constant over the entire domain. An artificially high value was used to keep the Reynolds number small for the computations that are illustrated in Figure 3. The visualization is based on the open source packages VTK [14], ParaView [15] and HiVision [16].

C. Augmented Reality Visualization

The visualization method is based on the accurate alignment of the viewer's position and the orientation of his camera view with the three-dimensional city model and the numerical simulation. Beside the accurate localization, the methodology of reality augmention is also of high relevance for the comprehensibility and credibility of the visualization. In the setup considered here only the graphics representing the flow field are to be embedded in the real-life image, and therefore the city model and the computational mesh should not be visible. Yet, the simulation results that are covered by buildings in the city model must be removed from the image. Therefore, the occluded simulation results are masked by the city model, which itself remains invisible leading to a masked visualization as displayed in Figure 4, where the transparent areas are left black.

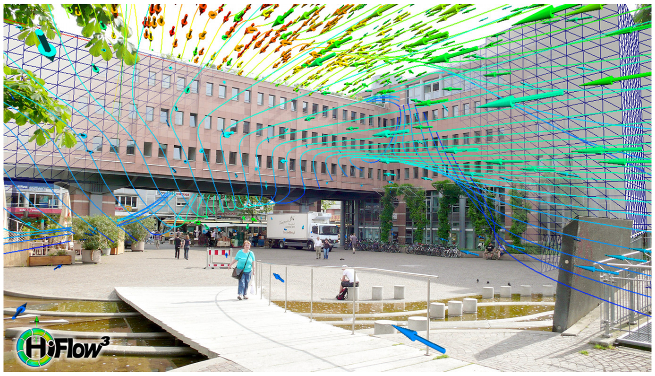


Figure 5. Enhanced augmented reality visualization

The masked visualization can then be composed onto the camera view leading to the augmented numerical simulation visualization in Figure 5, which was extended with the computational domain for illustration. The resulting image is very informative and gives insight into the simulation results. Since the displayed part of the simulation coincides with the viewer's position, the data selection is most intuitive and the full simulation can be explored by simply wandering around in the computational domain.

D. Interaction and User Interface

The interaction and the user interface is crucial for usability and comprehension. The proposed model is to present the mobile device as a window to the Augmented Reality and the results of the numerical simulation. This leads to challenges as outlined in [4] that can be addressed using sophisticated mathematical methods such as filtering, simulation and parameter identification. Only the increasing computing power available in modern mobile devices such as smartphones and tablets enable the use of such costly algorithms in real-time leading to haptic user interfaces.

The camera view in space is defined by six parameters,

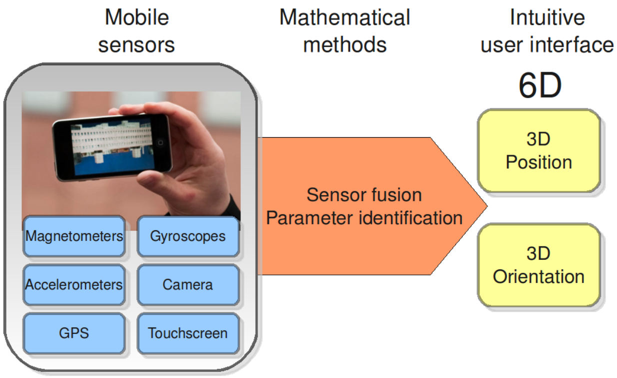


Figure 6. Mathematical methods enable intuitive user interfaces

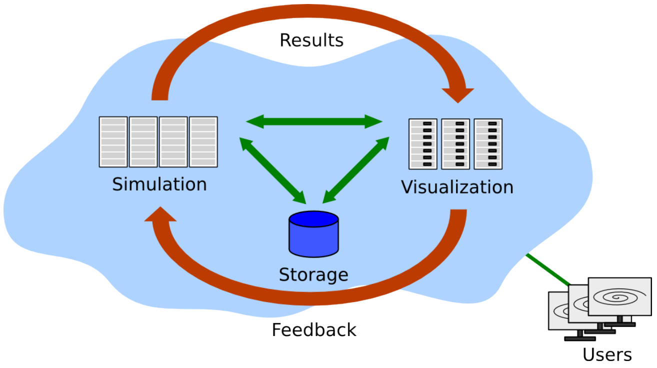


Figure 7. Interaction model

the three-dimensional position and the three viewing angles. Therefore at least six dimensions of sensor data is needed to control the user interface. Besides GPS, the latest generation of mobile devices contain spatial accelerometers as well as spatial magnetometers as a minimum. Taken together, they provide the six degrees of freedom in sensor data, enabling a new approach to an intuitive interface, which can be improved by any other additional sensors such as gyroscopes or camera based marker detection. Figure 6 illustrates that this step covers the real-time fusion of various sensor readings to gain the position and orientation information that is the basis for the Augmented Reality visualization.

Interaction with a numerical simulation consists not only of moving around and changing the view; it is highly desirable to also offer access to visualization parameters, such as what quantities are displayed, the method used, and potentially to enable changing some simulation parameters. From the view of the user interface, the touchscreen interfaces of modern mobile devices offer endless possibilities for manipulation of visualization and simulation parameters. Another crucial issue is the interactivity that is offered to the user: the presented visualization needs to be updated frequently, but is limited by the available network bandwidth.

E. Client-Server Framework

In general, the computation of large scale simulations and their visualizations need dedicated hardware and infrastructure, and is therefore traditionally only available to a small group of experts. The proposed visualization method overcomes this drawback by a client-server approach where the display, data selection and user interface is on a mobile device, but the actual simulation results and visualization remains on a high performance server infrastructure. The clients are connected to the visualization service on the servers by wireless or cellular networks as illustrated in Figure 7, which are limited by the available bandwidth. In a direct image transport a refresh rate of several frames per second is feasible on UMTS networks. Additional compression methods leading to higher refresh rates using significantly lower bandwidth are currently being developed by the authors in the framework of the newly starting European Project *MobileViz* in collaboration with industrial partners.

For widespread use of the simulation and visualization models, the necessary computing power can be provided by a cloud service, delivering the service of simulation and visualization on demand. The versatility and modularity of HiFlow^3 combined with the automated robust pre-processing of 3D city models and parallelized rendering servers for scientific visualization are the basis for a versatile and reliable service.

IV. Conclusion

In this paper, we have presented a novel visualization method for large-scale scientific computing illustrated by the example of urban air flow simulation. The use of mobile devices opens the path to intuitive access to and interaction with numerical simulations that are highly comprehensible due the embedding in to the real-life camera view as Augmented Reality visualizations. By this, results of numerical simulations will be available to decision-makers and citizens, raising the impact and improving the communication of scientific results. The presented methods are backed by a client-server framework and offer business models for simulation and visualization on demand in a cloud-based setup.

V. Acknowledgments

The *Karlsruhe Geometry* project is a collaboration of the Liegenschaftsamt of the city council of Karlsruhe with the Engineering Mathematics and Computing Lab (EMCL) and was supported by the KIT Competence Area for Information, Communication and Organization. The development of intuitive user interfaces for scientific applications on mobile devices is part of the *Science to Go* project, which received funding from the Apple Research & Technology Support (ARTS) programme.

References

[1] T. Kolbe, G. Gröger, and L. Plümer, "Citygml: Interoperable access to 3d city models," in *Geo-information for Disaster Management*, P. Oosterom, S. Zlatanova, and E. Fendel, Eds. Springer Berlin Heidelberg, 2005, pp. 883–899, 10.1007/3-540-27468-5_63.

[2] T. H. Kolbe, "Representing and exchanging 3d city models with citygml," in *Proceedings of the 3rd International Workshop on 3D Geo-Information, Lecture Notes in Geoinformation & Cartography*, J. Lee and S. Zlatanova, Eds. Seoul, Korea: Springer Verlag, 2009, p. 20.

[3] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment," in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*, oct 1997, pp. 74 –81.

[4] J. B. Gotow, K. Zienkiewicz, J. White, and D. C. Schmidt, "Addressing challenges with augmented reality applications on smartphones," in *MOBILWARE*, 2010, pp. 129–143.

[5] D. Schmalstieg, T. Langlotz, and M. Billinghurst, "Augmented reality 2.0," in *Virtual Realities*, G. Brunnett, S. Coquillart, and G. Welch, Eds. Springer Vienna, 2011, pp. 13–37, 10.1007/978-3-211-99178-7_2.

[6] D. Wagner, T. Pintaric, F. Ledermann, and D. Schmalstieg, "Towards massively multi-user augmented reality on handheld devices," in *In Third International Conference on Pervasive Computing*, 2005.

[7] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001.

[8] A. Nurminen, E. Kruijff, and E. E. Veas, "Hydrosys - a mixed reality platform for on-site visualization of environmental data," in *W2GIS*, 2010, pp. 159–175.

[9] H. Graf, P. Santos, and A. Stork, "Augmented reality framework supporting conceptual urban planning and enhancing the awareness for environmental impact," in *Proceedings of the 2010 Spring Simulation Multiconference*, ser. SpringSim '10. New York, NY, USA: ACM, 2010, pp. 181:1–181:8.

[10] M. J. Krause, "Fluid Flow Simulation and Optimisation with Lattice Boltzmann Methods on High Performance Computers: Application to the Human Respiratory System," Ph.D. dissertation, Karlsruhe Institute of Technology (KIT), Universität Karlsruhe (TH), Kaiserstraße 12, 76131 Karlsruhe, Germany, July 2010.

[11] K. Inthavong, J. Wen, J. Tu, and Z. Tian, "From CT Scans to CFD Modelling - Fluid and Heat Transfer in a Realistic Human Nasal Cavity," *Engineering Applications of Computational Fluid Mechanics*, vol. 3, no. 3, pp. 321–335, 2009.

[12] H. Anzt, W. Augustin, M. Baumann, H. Bockelmann, T. Gengenbach, T. Hahn, V. Heuveline, E. Ketelaer, D. Lukarski, A. Otzen, S. Ritterbusch, B. Rocker, S. Ronnås, M. Schick, C. Subramanian, J.-P. Weiss, and F. Wilhelm, "Hiflow3 - a flexible and hardware-aware parallel finite element package," in *Parallel/High-Performance Object-Oriented Scientific Computing (POOSC'10)*, accepted.

[13] T. Hauenstein, "Das 3D-Stadtmodell Karlsruhe," in *INTERGEO*, 2009. [Online]. Available: http://www.intergeo.de/archiv/2009/Hauenstein.pdf 29.7.2011

[14] "VTK - The Visualization Toolkit," http://www.vtk.org/ 29.7.2011.

[15] "ParaView - Open Source Scientific Visualization," http://www.paraview.org/ 29.7.2011.

[16] S. Bönisch and V. Heuveline, "Advanced flow visualization with hivision," in *Reactive Flows, Diffusion and Transport*, R. e. a. Rannacher, Ed. Springer, Berlin, 2006.

Parallel Computing of EFG Method on DRBL Cluster

Jiun-Yu Wu[12], Jiun Ting Lan[1], Kuen Tsann Chen[1], Yao-Tsung Wang[2], Steven Shiau[2],
[1]Department of Applied Mathematics, National Chung Hsing University, Taiwan
[2]National Center for High Performance Computing, Taiwan
e-mail: adherelinux@hotmail.com, lambow@pchome.com.tw, ktchen@amath.nchu.edu.tw, jazz@nchc.org.tw, steven@nchc.org.tw.

***Abstract*—In this paper, we present the application of element-free Galerkin (EFG) method to analyze two-dimensional elastostatics problems, for example, cantilever beam. MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method. We propose a powerful computation efficient architecture for CPU Cluster Using DRBL. The architecture help administrator to quickly deploy and manage CPU Cluster environment [15], it also bring benefit of computational efficiency in scientific computing. We have executed job on DBRL Cluster. The total time, speedup and efficiency have estimated for cantilever beam problem. We execute parallel implementation on DRBL Cluster [10]. For 16 cores, the speedup and efficiency are obtained to be 12.34 and 77.125% in cantilever beam problem.**

Keywords-EFG; parallel computing; DRBL Cluster; MPI.

I. Introduction

The partial differentiation equation usually describes a physical phenomenon in engineering science and engineering physics, like Navier-Stokes equation, heat conduction equation, vibration equation, wave equation, and so on. The smoothed particle hydrodynamics method (SPH) was proposed by Lucy [16] and it has been used to solve nonaxisymmetric phenomena in astrophysics. It has been applied to fluid mechanics and structure mechanics, etc. The SPH is represented by a set of particles, which move according to governing equations.

EFG can tackle initial value, boundary problems, linear and nonlinear partial differential problems [7-9]. Meshless methods, computational simulation techniques whose discrete model of the problem domain is described by nodes instead of predefined meshes [1-4]. Belytschko et al. [5, 6] (1994) developed the Element-Free Galerkin (EFG) method which used the moving least-squares (MLS) approximation to construct the shape function and employed Lagrange multipliers to satisfy the essential boundary condition. We use multiprocessor to calculate EFG numerical simulation. MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method.

MPICH2 is a tool of the Message-Passing Interface for CPU [11, 12]. MPICH2 was proposed by Argonne National Lab. MPICH2 is open source which is freely available license. It support system, including Microsoft Windows, Unix and Linux (ubuntu, centos, Fedora, etc.). The latest of version is 2-1.0 that we can download on official website.MPICH2 is implementation for distributed-memory and shared memory in parallel computing. MPICH2 offer parallel programming library which supports C, C++, Fortran language. The MPICH2 offers us some library which uses very convenience. In this paper, our program has been written in C language using MPI message passing library and execute on DRBL Cluster architecture. We use MPICH2 software easily by DRBL Cluster architecture.

II. MATHEMATICAL FORMULATION

A. Moving Least Squares Approximation

Moving least square (MLS) interpolants is used for the construction of the shape function in EFG method. The approximation $u^h(x)$ of the field variable $u(x)$ in the domain Ω has the following form:

$$\mathrm{u}^{\mathrm{h}}(\mathbf{x}) = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{1}$$

where

$$\mathbf{p}^{\mathbf{T}}(\mathbf{x}) = \begin{bmatrix} 1 & \mathrm{x} & \mathrm{y} & \mathrm{x}^2 & \mathrm{xy} & \mathrm{y}^2 \end{bmatrix}, \text{ for 2-D} \tag{2}$$

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \tag{3}$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{\mathrm{i}}^{\mathrm{n}} \mathrm{w}(\mathbf{x}\text{-}\mathbf{x}_{\mathrm{i}})\mathbf{p}^{\mathbf{T}}(\mathbf{x}_{\mathrm{i}})\mathbf{p}(\mathbf{x}_{\mathrm{i}}) \tag{4}$$

$$\mathbf{B}(\mathbf{x}) = \left[\mathrm{w}(\mathbf{x}\text{-}\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), \cdots, \mathrm{w}(\mathbf{x}\text{-}\mathbf{x}_{\mathrm{n}})\mathbf{p}(\mathbf{x}_{\mathrm{n}})\right] \tag{5}$$

$$\mathbf{u}^{\mathbf{T}} = [\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_{\mathrm{n}}] \tag{6}$$

In the present study, exponential weight function [1] was used as

$$\mathrm{w}(\mathrm{d}_{\mathrm{I}}^2) = \begin{cases} \dfrac{\mathrm{e}^{-(\mathrm{d}_{\mathrm{I}}/\mathrm{c})^2} - \mathrm{e}^{-(\mathrm{d}_{\mathrm{mI}}/\mathrm{c})^2}}{\left(1-\mathrm{e}^{-(\mathrm{d}_{\mathrm{mI}}/\mathrm{c})^2}\right)}, & \mathrm{d}_{\mathrm{I}} \le \mathrm{d}_{\mathrm{mI}} \\ 0, & \mathrm{d}_{\mathrm{I}} > \mathrm{d}_{\mathrm{mI}} \end{cases} \tag{7}$$

Hence, we have

$$u^{h}\left(\mathbf{x}\right)=\mathbf{\Phi}\left(\mathbf{x}\right)\mathbf{u} \tag{8}$$

B. *EFG Method with Lagrange Multipliers*

In the linear elastostatics problem, the variational form with Lagrange multipliers is given by [2]

$$\int_{\Omega}\delta\left(\mathbf{Lu}\right)^{T}\left(\mathbf{CLu}\right)d\Omega-\int_{\Omega}\delta\mathbf{u}^{T}\mathbf{b}d\Omega-\int_{\Gamma_t}\delta\mathbf{u}^{T}\bar{\mathbf{t}}d\Gamma$$
$$-\int_{\Gamma_u}\delta\lambda^{T}\left(\mathbf{u}-\bar{\mathbf{u}}\right)d\Gamma-\int_{\Gamma_u}\delta\mathbf{u}^{T}\lambda d\Gamma=0 \tag{9}$$

a where **L** is a matrix differential operator; **C** is a matrix of material constants; **b** is the vector of external body forces; λ is vector of the Lagrange multipliers.

Hence, the final discrete equation can be written in the following matrix form:

$$\begin{bmatrix}\mathbf{K} & \mathbf{G}\\ \mathbf{G}^{T} & \mathbf{0}\end{bmatrix}\begin{Bmatrix}\mathbf{u}\\ \lambda\end{Bmatrix}=\begin{Bmatrix}\mathbf{f}\\ \mathbf{q}\end{Bmatrix} \tag{10}$$

where

$$\mathbf{K}_{ij}=\int_{\Omega}\mathbf{D}_i^{T}\mathbf{C}\mathbf{D}_j d\Omega \tag{11}$$

$$\mathbf{D}_i=\mathbf{L}\mathbf{\Phi}_j=\begin{bmatrix}\Phi_{i,x} & 0\\ 0 & \Phi_{i,y}\\ \Phi_{i,y} & \Phi_{i,x}\end{bmatrix} \tag{12}$$

$$\mathbf{G}_{ij}=-\int_{\Gamma_u}\mathbf{N}_i^{T}\mathbf{\Phi}_j d\Gamma \tag{13}$$

$$\mathbf{D}=\frac{E}{1-\nu^{2}}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{(1-\nu)}{2}\end{bmatrix}\quad\text{for plane stress} \tag{14}$$

and $\mathbf{N}_i$ is the Lagrange interpolant for node *i* on the essential boundary.

III. System Environment

A. *DRBL*

Diskless Remote Boot in Linux (DRBL) is an open source solution to managing the deployment of the GNU/Linux operating system across many clients. DRBL supports lots of popular GNU/Linux distributions, and it is developed based on diskless and systemless environment for client machines. Figure 1 shows DRBL system architecture. DRBL uses PXE/Etherboot, DHCP, TFTP, NFS and NIS to provide services to client machines, so it is not necessary to install GNU/Linux on the client hard drives individually. Users just prepare a server machine for DRBL to be installed as a DRBL server, and follow the DRBL installation wizard to configure and dispose the environment for client machines step by step. It is really an easy job to deploy a DRBL environment on clustering systems even for a GNU/Linux beginner, hence cross-platform and user-friendly are the key factors that make the DRBL become a superior clustering tool.

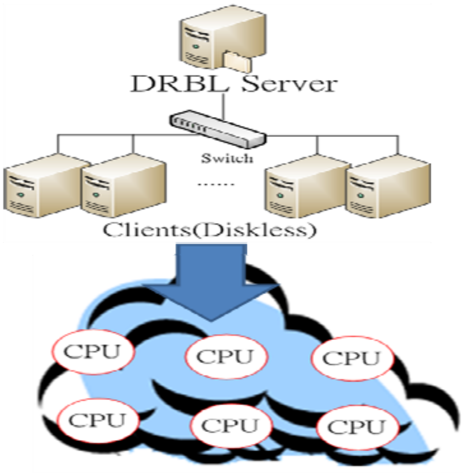


Figure 1. DRBL Cluster architecture.

DRBL can efficiently deploy diskless or diskfull cluster environment, and manage client. It configures these services (TFTP, NIS, DHCP, and NFS) to build a cluster environment. According to this implementation, administrator just needs two steps to deploy cluster environment. (1) Step 1: Installs DRBL packages and generates kernel and initrd for client; (2) Step 2: setup environment parameters, such IP address, and numbers of clients. It also provides cluster management and cluster system transformation (diskfull or diskless system).

The DRBL Cluster uses computers of PC classroom in our research center. One of computers has already been installed software (such as: MPICH2, fort77, g++, gcc, etc.) as DRBL server. It's very flexible to transform between two different modes cluster environment (diskfull and diskless) through DRBL. The cluster has 1 server、7 clients, The PC are equipped with Intel® Core(TM)2 Quad CPU Q9550 @ 2.83GHz. Table I illustrates Hardware specifications and Software list.

TABLE I. Hardware Specifications and Software List

Hardware (PC)	Software
Intel(a) Core™ 2 Quad CPU Q6600 2.4Hz	Ubuntu 10.04
8 GB RAM	Kernel 2.6.32.21

160GB Hard disk	DRBL 1.94-27
Intel 82571EB Gigabit NIC	gcc, g++, fort77, MPICH2
Hardware(Network switch)	
Linksys SLM2048 48 port 10/100 Gigabit Switch	

IV. NUMERICAL EXAMPLE AND PARALLEL IMPLEMENTATION

Consider a cantilever beam with length **L**=48*m* and height **W**=15 *m* was subjected to a concentrated load **P**=1000 *N* at the free end, as shown in Fig. 2.

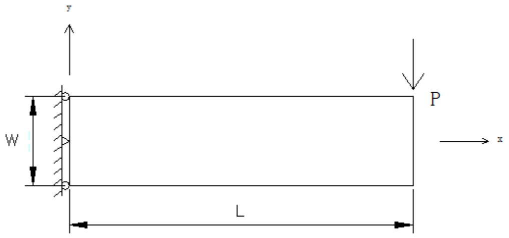


Figure 2. Cantilever Beam

This case was solved for the plane stress with Young's modulus $E = 3\times10^7$ *N/m* and Poisson's ratio $\nu = 0.3$. A regular arrangement of 48×15 nodes and regular integration cells with 4×4 Gauss quadrature were used. The normal stress σ_x and the shear stress τ_{xy} along the line *x*=L/2 is shown in Figs. 3 and 4, respectively. The relative error of the σ_{max} and τ_{max} of *x*=L/2 are 0.06% and 0.77%.

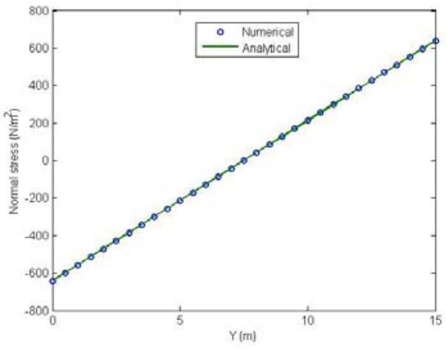


Figure 3. Distribution of normal stress σ_x at t x=L/2.

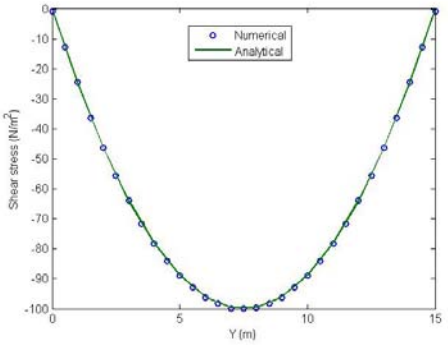


Figure 4. Distribution of Shear stress τ_{xy} at x=L/2.

A. Parallel algorithm

We have divided data decomposition in the parallel algorithm. Whole domain is divided into small subdomains and each processor performs the work on the subdomains. The EFG parallel implementation is consist of three parts in our code.

- First part is the following: Input data, nodal coordinates Gauss points, shape functions and its derivatives. Set up on the essential boundary. Calculation of each cell (subdomains for {k}).
- Secondary part is the following: Each processor sends subdomains to assembly of the system matrices {K}. We will obtain the formulation : AX=B
- Finally, A and B are known, we can use Gaussian Elimination for solving X unknowns.

We have measured performance between a multiprocessor system and a single processor system. Table II shows performance efficient. The figure 5 shows variation of speedup with number of processors. The figure 6 shows variation of efficiency with number of processors.

TABLE II. PERFORMANCE EFFICIENT

Number of processors	Total time	Speedup	Efficiency (%)
1	13608.5	1	100
4	3428.27	3.97	99.25
8	1759.8	7.73	96.625
16	1102.65	12.34	77.125

B. *Total time*

The total time is parallel computation run time. The total time is measured by MPI_Wtime which is MPI`s library functions.

C. *Speed up*

We obtained more great performance efficient by 16 processors. The speedup formulation is defined as following:

$$Speedup = \frac{ones\ processor(Total\ time)}{multi\ processors(Total\ time)}$$

D. *Parallel efficiency*

We have discussed about parallel efficiency here. The parallel efficiency formulation is defined as following:

$$parallel\ efficiency = \frac{Speedup}{Number\ of\ \ processors}$$

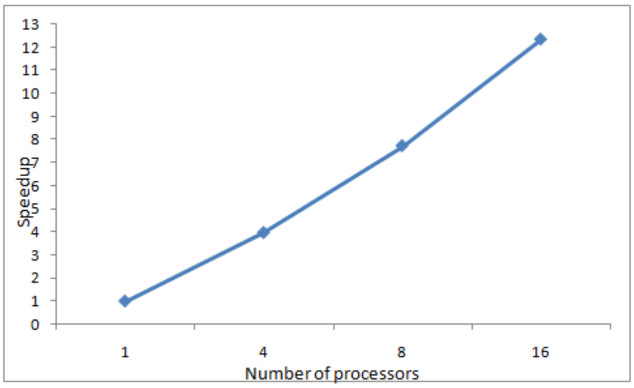


Figure 5. Variation of speedup with number of processors.

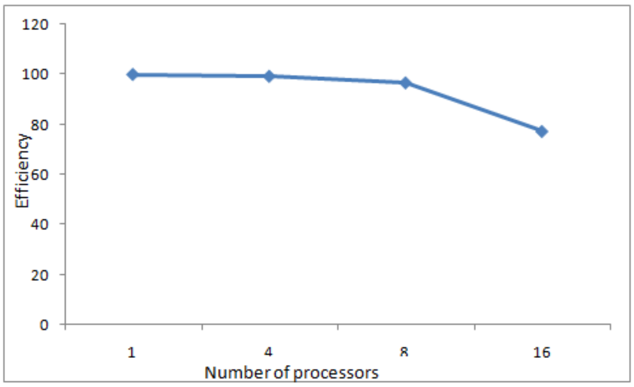


Figure 6. Variation of efficiency with number of processors.

V. Result Discussion

We have been developed parallel code in C language for cantilever beam problem. We have calculated for total time, speedup and efficiency on DRBL Cluster. The EFG method for cantilever beam problem is accuracy. In the future, we will use EFG method, to solve complex problem in engineering. We have performed parallel programming for EFG method. Our parallel programming has executed on DRBL Cluster system. The DRBL Cluster is very much useful in High performance computing environment.

We consider a cantilever beam that is excited by external force. We use EFG method to solve the normal stress and the shear stress along the line x=L/2. The relative error of normal stress and the shear stress of x=L/2 are 0.06% and 0.77%. The presented results confirm the theory for cantilever beam applications.

VI. Conclusion and Future Works

In the paper, we used EFG method to deal with an engineering problem. The finite element needs to build meshes. The EFG is no need to connect these nodes for whole domain. In other words, we don`t create and arrange the meshes. Therefore, EFG method just needs to build influence domain. EFG adopts a moving least square approximation function to fit based on nodes to make those field variables are continuous in the domain.

MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method.

In the future, we want to use EFG method to deal properly with kinds of complicate engineering problems like fracture extension, crack growth, kinematic boundary condition. We can employ GPGPU (General-Purpose Computing on Graphics Processing Unit) to deal with large scale problems.

References

[1] D. Broek, “Elementary Engineering Fracture Mechanics”, Noordhoff International Publishing, pp. 332, 1974.

[2] G.R. Liu and Y.T. Gu, “An Introduction to Meshfree Methods and Their Programming,” Springer, 2005.

[3] G.R. Liu,”Meshfree Methods Moving beyond the Finite Element Method,” CRC Press, USA. 2003.

[4] P. Lancaster, and K. Salkauskas, ”Surfaces by Generated Moving Least Squares Method,” Mathematics of Computation, vol. 37, pp. 141-158, 1981.

[5] T.Belytschko, Y.Y. Lu ,and L.Gu, “Element-free Galerkin Method”, Int. J. Numer. Method Eng, vol. 37, pp. 229-256, 1994.

[6] T. Belytschko, Y. Y. Lu ,and L. Gu, “Crack Propagation by element Free Galerkin Method,”, engineering fracture mechanics, vol. 51, no. 2, pp. 295-315, 1995.

[7] S Beissel and T. Belytschko, “Nodal integration of the element-free Galerkin method,” Comput. Methods. Appl. Mech. Eng, vol. 139, pp. 49-74, 1996.

[8] L. B. Lucy, “A Numerical Approach to the Testing of the Fission Hypothesis,” The Astron. J, vol. 8, pp. 1013-1024, 1977.

[9] B. Nayroles, G. Touzot and P. Villon, “Generalizing the finite element method diffuse approximation and diffuse elements,” Comput. Mech. vol. 10, pp. 307-318, 1992.

[10] W. C. Kuo, C. Y. Tu and Y. T. Wang, "Deploy Kerrighed SSI Massively Using DRBL," HPC ASIA 2009, 2009.

[11] DeinoMPI. [Online]. http://mpi.deino.net/ [accessed; Oct, 2011].

[12] MPICH2 [Online].

http://phase.hpcc.jp/mirrors/mpi/mpich2/index.htm. [accessed; Oct, 2011].

[13] C. Y. Tu, W. C. Kuo, Y. T. Wang and S. Shiau, "Building Energy Efficient ClassCloud using DRBL",10th IEEE/ACM International Conference Grid Computing.

[14] Diskless Remote Boot in Linux (DRBL), NCHC.

[Online]. http://drbl.sourceforge.net/ [accessed; Oct, 2011].

[15] J. Cope, M. Oberg, H. M. Tufo, and M. Woitaszek,"Shared Parallel Filesystems in Heterogeneous Linux Multi-Cluster Environments," proceedings of the 6th LCI International Conference on Linux Clusters: The HPC Revolution, 2005.

[16] L.B Lucy, “A numerical Approach to the Testing of the Fission Hypothesis”, The Atron. Astronomical Journal, vol. 82, pp. 1013-1024, 1977.

LZW versus Sliding Window Compression on a Distributed System: Robustness and Communication

Sergio De Agostino
Computer Science Department
Sapienza University
Rome, Italy
Email: deagostino@di.uniroma1.it

***Abstract*—Scalability preserves the robustness of sliding window compression only on very large files when it is implemented on a distributed system with low communication cost. On the other hand, we show that Lempel-Ziv-Welch compression is scalable and robust on arbitrary files.**

***Keywords*-dictionary-based compression, string factorization, parallel complexity, distributed algorithm.**

I. Introduction

Lempel-Ziv compression [1], [2], [3] is based on string factorization. Two different factorization processes exist with no memory constraints. With the first one (LZ1) [2], each factor is independent from the others since it extends by one character the longest match with a substring to its left in the input string (sliding window compression). With the second one (LZ2) [3], each factor is instead the extension by one character of the longest match with one of the previous factors (Lempel-Ziv-Welch or LZW compression). This computational difference implies that while sliding window compression has efficient parallel algorithms [4], [5] LZW compression is hard to parallelize [6]. This difference is mantained when bounded memory versions of Lempel-Ziv compression are considered [5], [7], [8]. On the other hand, parallel decompression is possible for both approaches [10]. This field has developed in the last twenty years from a theoretical approach concerning parallel time complexity with no memory constraints to the practical goal of designing distributed algorithms with bounded memory and low communication cost. While with shared memory machines scalability is always possible [11], this is not always garanteed with distributed memory. Distributed systems have two types of complexity, the interprocessor communication and the input-output mechanism. While the input/output issue is inherent to any parallel algorithm and has standard solutions, the communication cost of the computational phase after the distribution of the data among the processors and before the output of the final result is obviously algorithm-dependent. So, we need to limit the interprocessor communication and involve more local computation to design a practical algorithm. The simplest model for this phase is, of course, a simple array of processors with no interconnections and, therefore, no communication cost. An example of distibuted system with low communication cost is a tree architecture. Distributed algorithms for sliding window compression approximating in practice its compression effectiveness has been realized in [8] on an array of processor with no interprocessor communication. An approach using a tree architecture slightly improves compression effectiveness [9]. However, the scalability of a parallel implementation of sliding window compression on a distributed system with low communication cost garantees robustness only on very large files. On the other hand, we show in this paper that LZW compression is scalable and robust on arbitrary files if implemented on a tree architecture.

In Section 2, we describe the Lempel-Ziv compression techniques and in Section 3, we present the bounded memory versions. In Section 4, we present previous work ona parallel system with shared memory. Section 5 discuss how Lempel-Ziv data compression and decompression can be implemented on a distributes system and compare LZW compression with the sliding window technique. Conclusions and future work are given in Section 6.

II. Lempel-Ziv Data Compression

Lempel-Ziv compression is a dictionary-based technique. In fact, the factors of the string are substituted by *pointers* to copies stored in a dictionary, which are called *targets*. LZ1 (sliding window) compression is also called the sliding dictionary method while LZ2 (LZW) compression is more generally called the dynamic dictionary method.

A. Sliding Window Compression

Given an alphabet A and a string S in A^* the LZ1 factorization of S is $S = f_1 f_2 \cdots f_i \cdots f_k$ where f_i is the shortest substring, which does not occur previously in the prefix $f_1 f_2 \cdots f_i$ for $1 \leq i \leq k$. With such factorization, the encoding of each factor leaves one character uncompressed. To avoid this, a different factorization was introduced (LZSS factorization) where f_i is the longest match with a substring occurring in the prefix $f_1 f_2 \cdots f_i$ if $f_i \neq \lambda$, otherwise f_i is the alphabet character next to $f_1 f_2 \cdots f_{i-1}$ [12]. f_i is encoded by the pointer $q_i = (d_i, \ell_i)$, where d_i is the

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

displacement back to the copy of the factor and ℓ_i is the length of the factor (LZSS compression). If $d_i = 0$, l_i is the alphabet character. In other words the dictionary is defined by a window sliding its right end over the input string, that is, it comprises all the substrings of the prefix read so far in the computation. It follows that the dictionary is both *prefix* and *suffix* since all the prefixes and suffixes of a dictionary element are dictionary elements. The position of the longest match in the prefix with the current position can be computed in real time by means of a suffix tree data structure [13], [14].

B. LZW Compression

The LZ2 factorization of a string S is $S = f_1 f_2 \cdots f_i \cdots f_k$ where f_i is the shortest substring, which is different from one of the previous factors. As for LZ1 the encoding of each factor leaves one character uncompressed. To avoid this a different factorization was introduced (LZW factorization) where each factor f_i is the longest match with the concatenation of a previous factor and the next character [15]. f_i is encoded by a pointer q_i to such concatenation (LZW compression). LZW compression can be implemented in real time by storing the dictionary with a trie data structure. Differently from sliding window compression, the dictionary is only prefix.

C. Greedy versus Optimal Factorization

The pointer encoding the factor f_i has a size increasing with the index i. This means that the lower is the number of factors for a string of a given length the better is the compression. The factorizations described in the previous subsections are produced by greedy algorithms. The question is whether the greedy approach is always optimal, that is, if we relax the assumption that each factor is the longest match can we do better than greedy? The answer is negative with suffix dictionaries as for sliding window compression. On the other hand, the greedy approach is not optimal for LZW compression. However, the optimal approach is NP-complete [16] and the greedy algorithm approximates with an O$(n^{\frac{1}{4}})$ multiplicative factor the optimal solution [17].

III. Bounded Size Dictionary Compression

The factorization processes described in the previous section are such that the number of different factors (that is, the dictionary size) grows with the string length. In practical implementations instead the dictionary size is bounded by a constant and the pointers have equal size. While for sliding window compression this can be simply obtained by bounding the match and window lengths (therefore, the left end of the window slides as well), for LZW compression dictionary elements are removed by using a deletion heuristic. The deletion heuristics we describe in this section are FREEZE, RESTART and LRU [18]. Then, we give more details on sliding window compression.

A. The Deletion Heuristics

Let $d + \alpha$ be the cardinality of the fixed size dictionary where α is the cardinality of the alphabet. With the FREEZE deletion heuristic, there is a first phase of the factorization process where the dictionary is filled up and "freezed". Afterwards, the factorization continues in a "static" way using the factors of the freezed dictionary. In other words, the LZW factorization of a string S using the FREEZE deletion heuristic is $S = f_1 f_2 \cdots f_i \cdots f_k$ where f_i is the longest match with the concatenation of a previous factor f_j, with $j \leq d$, and the next character. The shortcoming of this heuristic is that after processing the string for a while the dictionary often becomes obsolete. A more sophisticated deletion heuristic is RESTART, which monitors the compression ratio achieved on the portion of the imput string read so far and, when it starst deteriorating, restarts the factorization process. Let $f_1 f_2 \cdots f_j \cdots f_i \cdots f_k$ be such factorization with j the highest index less than i where the restart operation happens. Then, f_j is an alphabet character and f_i is the longest match with the concatenation of a previous factor f_h, with $h \geq j$, and the next character (the restart operation removes all the elements from the dictionary but the alphabet characters). This heuristic is used by the Unix command Compress since it has a good compression effectiveness and it is easy to implement. However, the best deletion heuristic is LRU (last recently used strategy). The LRU deletion heuristic removes elements from the dictionay in a continuous way by deleting at each step of the factorization the least recently used factor, which is not a proper prefix of another one.

B. Compression with Finite Windows

As mentioned at the beginning of this section, bounded size dictionary compression can also be obtained by sliding a fixed length window and by bounding the match length. A real time implementation of compression with finite window is possible using a suffix tree data structure [19]. Much simpler real time implementations are realized by means of hashing techniques providing a specific position in the window where a good appriximation of the longest match is found on realistic data. In [20], the three current characters are hashed to yield a pointer into the already compressed text. In [21], hashing of strings of all lengths is used to find a match. In both methods, collisions are resolved by overwriting. In [22], the two current characters are hashed and collisions are chained via an offset array. Also the Unix gzip compressor chains collisions but hashes three characters [23].

C. Greedy versus Optimal Factorization

Greedy factorization is optimal for compression with finite windows since the dictionary is suffix. With LZW compression, after we fill up the dictionary using the FREEZE or RESTART heuristic, the greedy factorization we compute with such dictionary is not optimal since the dictionary is not suffix. However, there is an optimal semi-greedy factorization, which at each step computes a factor such that the longest match in the next position with a dictionary element ends to the rightest [24], [25]. Since the dictionary is prefix, the factorization is optimal. The algorithm can even be implemented in real time with a modified suffix tree data structure [24].

IV. Previous Work

Sliding window compression can be efficiently parallelized on a PRAM EREW [4], [5], [8], that is, a parallel machine where processors access a shared memory without reading and writing conflicts. On the other hand, LZW compression is P-complete [6] and, therefore, hard to parallelize. Decompression, instead, is parallelizable for both methods [10]. As far as bounded size dictionary compression is concerned, the "parallel computation thesis" claims that sequential work space and parallel running time have the same order of magnitude giving theoretical underpinning to the realization of parallel algorithms for LZW compression using a deletion heuristic. However, the thesis concerns unbounded parallelism and a practical requirement for the design of a parallel algorithm is a limited number of processors. A stronger statement is that sequential logarithmic work space corresponds to parallel logarithmic running time with a polynomial number of processors. Therefore, a fixed size dictionary implies a parallel algorithm for LZW compression satisfying these constraints. Realistically, the satisfaction of these requirements is a necessary but not a sufficient condition for a practical parallel algorithm since the number of processors should be linear, which does not seem possible for the RESTART deletion heuristic. Moreover, the SC^k-completeness of LZ2 compression using the LRU deletion heuristic and a dictionary of polylogarithmic size shows that it is unlikely to have a parallel complexity involving reasonable multiplicative constants [7]. In conclusion, the only practical LZW compression algorithm for a shared memory parallel system is the one using the FREEZE deletion heuristic. We will see these arguments more in details in the next subsections.

A. Sliding Window Compression on a Parallel System

We present compression algorithms for sliding dictionaries on an exclusive read, exclusive write shared memory machine requiring $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors if k is $\Omega(\log n)$, with the practical and realistic assumption that the dictionary size and the match length are constant [8]. As previously mentioned, greedy factorization is optimal with sliding dictionaries. In order to compute a greedy factorization in parallel we find the greedy match in each position i of the input string and link i to $j+1$, where j is the last position of the match. If the greedy match ends the string i is linked to $n+1$, where n is the length of the string. It follows that we obtain a tree rooted in $n+1$ and the positions of the factors are given by the path from 1 to $n+1$. Such tree can be built in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors. In fact, on each block of k positions one processor has to compute a match having constant length and the reading conflicts with other processors are solved in logarithmic time by standard broadcasting techniques. Then, since for each node of the tree the number of children is bounded by the constant match length it is easy to add the links from a parent node to its children in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors and apply the well-known Euler tour technique to this doubly linked tree structure to compute the path from 1 to $n+1$.

B. The Completeness Results

NC is the class of problems solvable with a polynomial number of processors in polylogarithmic time on a parallel random access machine and it is comjectured to be a proper subset of P, the class of problems solvable in sequential polynomial time. LZ2 and LZW compression with an unbounded dictionary have been proved to be P-complete [6] and, therefore, hard to parallelize. SC is the class of problems solvable in polylogarithmic space and sequential polynomial time. The LZ2 algorithm with LRU deletion heuristic on a dictionary of size $\mathrm{O}(\log^k n)$ can be performed in polynomial time and $\mathrm{O}(\log^k n \log\log n)$ space, where n is the length of the input string. In fact, the trie requires $\mathrm{O}(\log^k n)$ space by using an array implementation since the number of children for each node is bounded by the alphabet cardinality. The $\log\log n$ factor is required to store the information needed for the LRU deletion heuristic since each node must have a different age, which is an integer value between 0 and the dictionary size. Obviously, this is true for the LZW algorithm, as well. If the size of the dictionary is $\mathrm{O}(\log^k n)$, the LRU strategy is log-space hard for SC^k, the class of problems solvable simultaneously in polynomial time and $\mathrm{O}(\log^k n)$ space [7]. The problem belongs to SC^{k+1}. This hardness result is not so relevant for the space complexity analysis since $\Omega(\log^k n)$ is an obvious lower bound to the work space needed for the computation. Much more interesting is what can be said about the parallel complexity analysis. In [7], it was shown that LZ2 (or LZW) compression using the LRU deletion heuristic with a dictionary of size c can be performed in parallel either in $\mathrm{O}(\log n)$ time with $2^{\mathrm{O}(c \log c)} n$ processors or in $2^{\mathrm{O}(c \log c)} \log n$ time with $\mathrm{O}(n)$ processors. This means that if the dictionary size is constant, the compression problem belongs to NC. NC and SC are classes that can be viewed in some sense symmetric and are believed to be incomparable. Since log-space reductions are in NC, the compression problem cannot belong to NC when the dictionary size is polylogarithmic if NC and SC are incomparable. We want to point out that the dictionary size c figures as an exponent in the parallel complexity of the problem. This is not by accident. If we believe that SC is not included in NC, then the SC^k-

hardness of the problem when c is $\mathrm{O}(\log^k n)$ implies the exponentiation of some increasing and diverging function of c. In fact, without such exponentiation either in the number of processors or in the parallel running time, the problem would be SC^k-hard and in NC when c is $\mathrm{O}(\log^k n)$. Observe that the P-completeness of the problem, which requires a superpolylogarithmic value for c, does not suffice to infer this exponentiation since c can figure as a multiplicative factor of the time function. Moreover, this is a unique case so far where somehow we use hardness results to argue that practical algorithms of a certain kind (NC in this case) do not exist because of huge multiplicative constant factors occurring in their analysis. In [7], a relaxed version (RLRU) was introduced, which turned out to be the first (and only so far) natural SC^k-complete problem.

C. LZW Compression on a Parallel System

As mentioned at the beginning of this section, the only practical LZW compression algorithm for a shared memory parallel system is the one using the FREEZE deletion heuristic. After the dictionary is built and freezed, a parallel procedure similar to the one for sliding window compression is run. To compute a greedy factorization in parallel we find the greedy match with the freezed dictionary in each position i of the input string and link i to $j+1$, where j is the last position of the match. If the greedy match ends the string i is linked to $n+1$, where n is the length of the string. It follows that we obtain a tree rooted in $n+1$ and the positions of the factors of the greedy parsing are given by the path from 1 to $n+1$. In order to compute an optimal factorization we parallelize the semi-greedy procedure. The longest sequence of two matches in each position i of the string can be computed in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors, in a similar way as for the greedy procedure. Then, position i is linked to the position of the second match. If the second match is empty, i is linked to $n+1$. Again, we obtain a tree rooted in $n+1$ and the positions of the factors are given by the path from 1 to $n+1$. The tree and the path are computed in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors if k is $\Omega(\log n)$, as in the first subsection without reading and writing conflicts [8]. The parallelization of the sequential LZW compression algorithm with the RESTART deletion heuristic is not practical enough since it requires a quadratic number of processors [7].

D. Parallel Deompression

The design of parallel decoders is based on the fact that the Euler tour technique can also be used to find the trees of a forest in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors on a shared memory parallel machine without writing and reading conflicts, if k is $\Omega(\log n)$ and n is the number of nodes. We present decoders paired with the practical coding implementations using bounded size dictionaries. First, we see how to decode the sequence of pointers $q_i = (d_i, \ell_i)$ produced by the sliding window method with $1 \leq i \leq m$ [10]. If $s_1, ..., s_m$ are the partial sums of $l_1, ..., l_m$, the target of q_i encodes the substring over the positions $s_{i-1}+1 \cdots s_i$ of the output string. Link the positions $s_{i-1}+1 \cdots s_i$ to the positions $s_{i-1}+1-d_i \cdots s_{i-1}+1-d_i+l_i-1$, respectively. If $d_i = 0$, the target of q_i is an alphabet character and the corresponding position in the output string is not linked to anything. Therefore, we obtain a forest where all the nodes in a tree correspond to positions of the decoded string where the character is represented by the root. The reduction from the decoding problem to the problem of finding the trees in a forest can be computed in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors where n is the length of the output string, because this is the complexity of computing the partial sums since $m \leq n$. Afterwards, one processor stores the parent pointers in an array of size n for a block of k positions. We can make the forest a doubly linked structure since the window size is constant and apply the Euler tour technique to find the trees. With LZW compression using the FREEZE deletion heuristic the parallel decoder is trivial. We wish to point out that the decoding problem is interesting independently from the computational efficiency of the encoder. In fact, in the case of compressed files stored in a ROM only the computational efficiency of decompression is relevant. With the RESTART deletion heuristic, a special mark occurs in the sequence of pointers each time the dictionary is cleared out so that the decoder does not have to monitor the compression ratio. The positions of the special mark are detected by parallel prefix. Each subsequence $q_1 \cdots q_m$ of pointers between two consecutive marks can be decoded in parallel but the pointers do not contain the information on the length of their targets and it has to be computed. The target of the pointer q_i in the subsequence is the concatenation of the target of the pointer in position $q_i - \alpha$ with the first character of the target of the pointer in position $q_i - \alpha + 1$, where α is the alphabet cardinality. Then, in parallel for each i, link pointer q_i to the pointer in position $q_i - \alpha$, if $q_i > \alpha$. Again, we obtain a forest where each tree is rooted in a pointer representing an alphabet character and the length l_i of the target of a pointer q_i is equal to the level of the pointer in the tree plus 1. It is known from [1] that the largest number of distinct factors whose concatenation forms a given string of length ℓ is $\mathrm{O}(\ell/\log \ell)$. Since a factor of the LZW factorization of a string appears a number of times, which is at most equal to the alphabet cardinality, it follows that m is $\mathrm{O}(\ell/\log \ell)$ if ℓ is the length of the substring encoded by the subsequence $q_1 \cdots q_m$. Then, building such a forest takes $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$ processors on a shared memory parallel machine without writing and reading conflicts if k is $\Omega(\log n)$. By means of the Euler tour technique, we can compute the trees of such forest and the level of each node in its own tree in $\mathrm{O}(k)$ time with $\mathrm{O}(n/k)$. Therefore, we can compute the lengths $l_1, ..., l_m$ of the targets. If $s_1, ..., s_m$ are the partial sums, the target of q_i is the substring over the positions $s_{i-1}+1 \cdots s_i$ of the output string. For each q_i, which does not correspond to an alphabet character, define $first(i) = s_{q_i-\alpha-1} + 1$ and $last(i) = s_{q_i-\alpha} + 1$. Since the target of the pointer q_i is the concatenation of the target of the pointer in position $q_i - \alpha$ with the first character of the target of the pointer in position $q_i - \alpha + 1$, link the positions $s_{i-1} + 1 \cdots s_i$ to the positions $s_{first(i)} \cdots s_{last(i)}$, respectively. As in the sliding dictionary case, if the target of q_i is an alphabet character the corresponding position in the output string is the root of a tree in a forest and all the nodes in a tree correspond to positions of the decoded string where the character is the root. Since the number of children for each node is at most α, in O(k) time and O(n/k) processors we can store the forest in a doubly linked structure and decode by means of the Euler tour technique [10].

V. LZW versus Sliding Window Compression on a Distributed System

As mentioned in the introduction, the simplest distributed system is an array of processors with no interconnections. For every integer k greater than 1 an O(kw) time, O(n/kw) processors distributed algorithm factorizing an input string S with a cost, which approximates the cost of the LZSS factorization within the multiplicative factor $(k+m-1)/k$, where n, m and w are the lengths of the input string, the longest factor and the window respectively was presented on such model in [8]. As far as LZW compression is concerned, if we use a RESTART deletion heuristic clearing out the dictionary every ℓ characters of the input string we can trivially parallelize the factorization process with an O(ℓ) time, O(n/ℓ) processors distributed algorithm. In this paper we present on a tree architecture an algorithm, which in time O(km) with O(n/km) processors is guaranteed to produce a factorization of S with a cost approximating the cost of the optimal factorization within the multiplicative factor $(k+1)/k$. All the algorithms mentioned above provide approximation schemes for the corresponding factorization problems since the multiplicative approximation factors converge to 1 when km and kw converge to ℓ and to n, respectively.

A. Sliding Window Compression on a Distributed System

We simply apply in parallel sliding window compression to blocks of length kw. It follows that the algorithm requires O(kw) time with n/kw processors and the multiplicative approximation factor is $(k+m-1))/k$ with respect to any parsing. In fact, the number of factors of an optimal (greedy) factorization on a block is at least kw/m while the number of factors of the factorization produced by the scheme is at most $(k-1)w/m+w$. The boundary might cut a factor and the length w of the initial full size window of the block is the upper bound to the factors produced by the scheme in it. Yet, the factor cut by the boundary might be followed by another factor, which covers the remaining part of the initial window. If this second factor has a suffix to the right of the window, this suffix must be a factor of the sliding dictionary defined by it and the multiplicative approximation factor follows. We obtain an approximation scheme, which is suitable for a small scale system but due to its adaptiveness it works on a large scale parallel system when the file size is large. From a practical point of view, we can apply something like the gzip procedure to a small number of input data blocks achieving a satisfying degree of compression effectiveness and obtaining the expected speed-up on a real parallel machine. Making the order of magnitude of the block length greater than the one of the window length largely beats the worst case bound on realistic data and garantees robustness. The window length is usually several thousands kilobytes. The compression tools of the Zip family, as the Unix command “gzip” for example, use a window size of at least 32K. It follows that the block length in our parallel implementation should be about 300K and the file size should be about one third of the number of processors in megabytes.

B. LZW Compression on a Distributed System

As mentioned at the beginning of this section, if we use a RESTART deletion heuristic clearing out the dictionary every ℓ characters of the input string we can trivially parallelize the factorization process with an O(ℓ) time, O(n/ℓ) processors distributed algorithm. LZW compression with the RESTART deletion heuristic was initially presented in [15] with a dictionary of size 2^{12} and is employed by the Unix command “compress” with a dictionary of size 2^{16}. Therefore, in order to have a satisfying compression effectiveness the distributed algorithm might work with blocks of length ℓ even greater than 300K on realistic data. After a dictionary is filled up for each block though, the factorization of the remaining suffix of the block can be approximated within the multiplicative factor $(k+1)/k$ in time O(km) with O(n/km) processors on a tree architecture. Every leaf processor stores a sub-block of length $m(k+2)$ and a copy of the dictionary, which are broadcasted from some level of the tree where the first phase of the computation has been executed. Adjacent sub-blocks overlap on $2m$ characters. We call a *boundary match* a factor covering positions of two adjacent sub-blocks. We execute the following algorithm:

- for each block, every processor but the one associated with the last sub-block computes the boundary match between its sub-block and the next one, which ends furthest to the right;
- each processor computes the optimal factorization from the beginning of the boundary match on the left boundary of its sub-block to the beginning of the boundary match on the right boundary.

Stopping the factorization of each sub-block at the beginning of the right boundary match might cause the making of a surplus factor, which determines the multiplicative approximation factor $(k+1)/k$ with respect to any factorization. In

ISBN: 978-1-61208-161-8

fact, the factor in front of the right boundary match might be extended to be a boundary match itself and to cover the first position of the factor after the boundary. In [26], it is shown experimentally that for $k = 10$ the compression ratio achieved by such factorizarion is about the same as the sequential one. Then, compression is effective and robust on a large scale system even if the size of the file is not large.

C. Decompression on a Distributed System

To decode the compressed files on a distibuted system, it is enough to use a special mark occuring in the sequence of pointers each time the coding of a block ends. The input phase distributes the subsequences of pointers coding each block among the processors. If the file is encoded by an LZW compressor implemented on a large scale tree architecture, a second special mark indicates for each block the end of the coding of a sub-block and the coding of each block is stored at the same level of the tree. The first sub-block for each block is decoded by one processor to learn the corresponding dictionary. Then, the subsequences of pointers coding the sub-blocks are broadcasted to the leaf processors with the corresponding dictionary.

VI. Conclusion

In this paper, we showed that with the low communication cost of a tree architecture we can scale up the implementation of LZW compression on a distributed system preserving its robustness. This does not seem to be possible with sliding window compression. As future work, we would like to implement Lempel-Ziv compression on available distributed systems as array and tree architectures.

References

[1] A. Lempel and J. Ziv, *On the Complexity of Finite Sequences,* IEEE Transactions on Information Theory 22, 75-81, 1976.

[2] A. Lempel and J. Ziv, *A Universal Algorithm for Sequential Data Compression,* IEEE Transactions on Information Theory 23, 337-343, 1977.

[3] J. Ziv and A. Lempel, *Compression of Individual Sequences via Variable-Rate Coding,* IEEE Transactions on Information Theory 24, 530-536, 1978.

[4] M. Crochemore and W. Rytter, *Efficient Parallel Algorithms to Test Square-freeness and Factorize Strings,* Information Processing Letters 38, 57-60, 1991.

[5] S. De Agostino, *Parallelism and Dictionary-Based Data Compression,* Information Sciences 135, 43-56, 2001.

[6] S. De Agostino, *P-complete Problems in Data Compression,* Theoretical Computer Science 127, 181-186, 1994.

[7] S. De Agostino and R. Silvestri, *Bounded Size Dictionary Compression:* SC^k*-Completeness and* NC *Algorithms,* Information and Computation 180, 101-112, 2003.

[8] L. Cinque, S. De Agostino, and L. Lombardi, *Scalability and Communication in Parallel Low-Complexity Lossless Compression,* Mathematics in Computer Science 3, 391-406, 2010.

[9] S. T. Klein and Y. Wiseman, *Parallel Lempel-Ziv Coding,* Discrete Applied Mathematics 146, 180-191, 2005.

[10] S. De Agostino, *Almost Work-Optimal PRAM EREW Decoders of LZ-Compressed Text,* Parallel Processing Letters 14, 351-359, 2004.

[11] R. P. Brent, *The Parallel Evaluation of General Arithmetic Expressions,* Journal of the ACM 21, 201-206, 1974.

[12] J. A. Storer and T. G. Szimansky, *Data Compression via Textual Substitution,* Journal of ACM 24, 928-951, 1982.

[13] M. Rodeh, V. R. Pratt, and S. Even, *Linear Algorithms for Compression via String Matching,* Journal of ACM 28, 16-24, 1980.

[14] E. M. Mc Creight, *A Space-Economical Suffix Tree Construction Algorithm,* Journal of ACM 23, 262-272, 1976.

[15] T. A. Welch, *A Technique for High-Performance Data Compression,* IEEE Computer 17, 8-19, 1984.

[16] S. De Agostino and J. A. Storer, *On-Line versus Off-line Computation for Dynamic Text Compression,* Information Processing Letters 59, 169-174, 1996.

[17] S. De Agostino and R. Silvestri, *A Worst Case Analisys of the LZ2 Compression Algorithm,* Information and Computation 139, 258-268, 1997.

[18] J. A. Storer, *Data Compression: Methods and Theory,* Computer Science Press, 1988.

[19] E. R. Fiala and D. H. Green, *Data Compression with Finite Windows,* Communications of ACM 32, 490-505, 1988.

[20] J. R. Waterworth, *Data Compression System,* US Patent 4 701 745, 1987.

[21] R. P. Brent, *A Linear Algorithm for Data Compression,* Australian Computer Journal 19, 64-68, 1987.

[22] D. A. Whiting, G. A. George, and G. E. Ivey, *Data Compression Apparatus and Method,* US Patent 5016009, 1991.

[23] J. Gailly and M. Adler, http://www.gzip.org, 1991.

[24] A. Hartman and M. Rodeh, *Optimal Parsing of Strings.* In: Apostolico, A., Galil, Z. (eds.) Combinatorial Algorithms on Words, 155-167, Springer, 1985.

[25] M. Crochemore and W. Rytter, *Jewels of Stringology,* World Scientific, 2003.

[26] D. Belinskaya, S. De Agostino, and J. A. Storer, *Near Optimal Compression with respect to a Static Dictionary on a Practical Massively Parallel Architecture,* Proceedings IEEE Data Compression Conference, 172-181, 1995.

Self-Organizing the Selection of Migratable Processes on Cluster-of-Clusters Environments

Rodrigo da Rosa Righi, Lucas Graebin, Rafael Bohrer Ávila
Programa Interdisciplinar de Pós-Graduação em Computação Aplicada – UNISINOS – São Leopoldo - Brazil
Email: rrrighi@unisinos.br, lgraebin@acm.com, rbavila@unisinos.br

Philippe Olivier Alexandre Navaux, Laércio Lima Pilla
Programa De Pós-Graduação em Computação – UFRGS – Porto Alegre - Brazil
Email: {navaux, llpilla}@inf.ufrgs.br

***Abstract*—The decision to move processes to new resources is NP-Hard, and heuristics take place in order to reach good results inside an acceptable time interval. In this way, this paper presents AutoMig — a novel heuristic for BSP applications that self-organizes the selection of candidates for migration on different clusters. Its differential approach consists in a prediction function (*pf*) that considers both processes' computation and communication data as well as their migration costs. *pf* is applied over a list of schedules and AutoMig's final step decides whether one of them outperforms the time of the current mapping. The results emphasize gains up to 32% when testing a CPU-bound application in a simulated cluster-of-clusters environment. Besides AutoMig, this paper also describes the rescheduling model associated with it.**

***Keywords*-BSP, rescheduling, heuristic, self-organizing.**

I. Introduction

Generally, process migration is implemented within the application with explicit calls [11]. A different migration approach happens at middleware level, where changes in the application code and previous knowledge about the system are usually not required. Considering this, we have developed a process rescheduling model called MigBSP [4]. It was designed to work with round-based applications with a BSP behavior (Bulk Synchronous Parallel). Concerning the choosing of the processes, MigBSP creates a priority list based on the highest Potential of Migration (*PM*) of each process [4]. *PM* combines the migration costs with data from computation and communication phases in order to create an unified scheduling metric. The process denoted on the top of the list is selected to be inspected for migration. Although we achieved good results with this approach, we agree that a selection of a percentage of processes could determine better results. However, a question arises: How can one reach an optimized value for dynamic environments? A solution involves the testing of several hand-tuned parameters and a comparison among the results.

After developing the first version of MigBSP, we have observed the promotion of intelligent scheduling systems which adjust their parameters on the fly and hide intrinsic optimization decisions from users [11]. In this context, we developed a new heuristic named **AutoMig** that selects one or more candidates for migration automatically. We took advantage of both List Scheduling and Backtracking concepts to evaluate the migration impact on each element of the *PM* list in an autonomous fashion. In addition, another AutoMig's strength comprises the needlessness to provide an additional parameter in MigBSP for getting more than one migratable process on rescheduling activation. The scheduling evaluation uses a prediction function (*pf*) that considers the migration costs and works following the concept of a BSP superstep [1]. The lowest pf indicates the most suitable rescheduling plan.

This paper aims to describe AutoMig in details. We evaluated it by using an BSP application for image compression [7]. Considering that the programmer does not need to change his/her application nor add a parameter in MigBSP, the results with migration were satisfactory and totaled a mean gain of 7.9%. This index was observed when comparing migrations with the application execution solely. The results also showed a serie of situations where AutoMig outperforms the heuristic that elects only one process.

We organized the paper in eight sections. Section 2 presents MigBSP. The main part of the paper belongs to Section 3, where Automig is described in details. Sections 4 and 5 show the employed methodology and the results, respectively. Related work is discussed in Section 6, while Section 7 presents the conclusion and future work. Finally, Section 8 shows our acknowledgments to Brazilian agencies.

II. MigBSP: Rescheduling Model

MigBSP is a rescheduling model that works over heterogeneous resources, joining the power of clusters, supercomputers and local networks. The heterogeneity issue considers the processors' clock (all processors have the same set of instructions), as well as network bandwidth. Such an architecture is assembled with Sets (sites) and Set Managers. Set Managers are responsible for scheduling, capturing data from a Set and exchanging it among other managers.

The decision for process remapping is taken at the end of a superstep. Aiming to generate the least intrusiveness in application as possible, we applied two adaptations that control the value of α ($\alpha \in \mathbb{N}^*$). α is updated at each rescheduling call and will indicate the interval for the next one. The adaptations' objectives are: (i) to postpone the rescheduling call if the processes are balanced or to turn it more frequent, otherwise; (ii) to delay this call if a pattern without migrations on ω past calls is observed. A variable named D is used to indicate a percentage of how far the slowest and the fastest processes may be from the average to consider the processes as balanced.

The answer for "Which" is solved through our decision function called Potential of Migration (*PM*). Each process i computes n functions $PM(i,j)$, where n is the number of Sets and j means a Set. The key rationale consists in performing only a subset of the processes-resources tests at the rescheduling moment. $PM(i,j)$ is found using Computation, Communication and Memory metrics as we can see in Equations 1, 2, 3 and 4. A previous paper describes each equation in details [4]. The greater the value of $PM(i,j)$, the more prone the processes will be to migrate.

$$Comp(i,j) = P_{comp}(i)\ .\ CTP(i)\ .\ ISet(j) \qquad (1)$$

$$Comm(i,j) = P_{comm}(i,j)\ .\ BTP(i,j) \qquad (2)$$

$$Mem(i,j) = M(i)\ .\ T(i,j)\ +\ Mig(i,j) \qquad (3)$$

$$PM(i,j) = Comp(i,j) + Comm(i,j) - Mem(i,j) \qquad (4)$$

Computation metric - $Comp(i,j)$ - considers a Computation Pattern $P_{comp}(i)$ that measures the stability of a process i regarding the amount of instructions at each superstep. This value is close to 1 if the process is regular and close to 0 otherwise. This metric also performs a computation time prediction $CTP(i)$ for process i based on all computation phases between two rescheduling activations. $Comp(i,j)$ also presents an index $ISet(j)$ which informs the average capacity of Set j. In the same way, Communication metric – $Comm(i,j)$ – computes the Communication Pattern $P_{comm}(i,j)$ between processes and Sets. Furthermore, this metric uses communication time prediction $BTP(i,j)$ considering data between two rebalancing activations. $Comm(i,j)$ increases if process i has a regular communication with processes from Set j and performs slower communication actions to this Set. Memory metric – $Mem(i,j)$ – considers process memory, transferring rate between considered process and the manager of target Set, as well as migration costs. These costs are dependent of the operating system, as well as the migration tool.

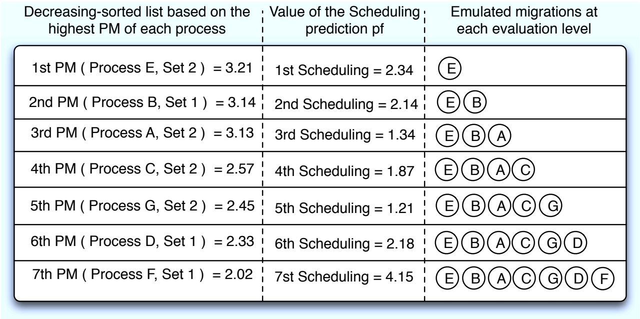
At each rescheduling call, each process passes its highest $PM(i,j)$ to its Set Manager. This last entity exchanges the *PM* of the processes with other managers. Each manager creates a decreasing-sorted list and selects the process on the top for testing the migration viability. This test considers the following data: (i) the external load on source and destination processors; (ii) the processes that both processors are executing; (iii) the simulation of considered process running on a destination processor; (iv) the time of communication actions considering local and destination processors; (v) migration costs. We computed two times: t_1 and t_2. t_1 means the local execution of process i, while t_2 encompasses its execution on the other processor and includes the migration costs. A new resource is chosen if $t_1 > t_2$.

III. AutoMig: A Novel Heuristic to Select the Suitable Processes for Migration

AutoMig's self-organizes the migratable processes without programmer intervention. It can elect not only one but a collection of processes at the migration moment. Especially, AutoMig's proposal solves the problem described below.

- **Problem Statement** - Given n BSP processes and a list of the highest *PM* of each one at the migration moment, the challenge consists in creating and evaluating at maximum n new scheduling plans and to choose the most profitable one among those that outperform the current processes-resources mapping.

AutoMig solves this question by using the concepts from List Scheduling and Backtracking. Firstly, we sort the *PM* list in a decreasing-ordered manner. Thus, the tests begin by the process on the head since its rescheduling represents better chances of migration gains. Secondly, AutoMig proposes n scheduling attempts (where n is the number of processes) by incrementing the movement of only one process at each new plan. This idea is based on the Backtracking functioning, where each partial candidate is the parent of candidates that differ from it by a single extension step. Figure 1 depicts an example of this approach, where a single migration on level l causes an impact on $l+1$. For instance, the performance forecast for process "A" considers its own migration and the fact that "E" and "B" were migrated too. Algorithm 1 presents AutoMig's approach in details.



Decreasing-sorted list based on the highest PM of each process	Value of the Scheduling prediction pf	Emulated migrations at each evaluation level
1st PM (Process E, Set 2) = 3.21	1st Scheduling = 2.34	E
2nd PM (Process B, Set 1) = 3.14	2nd Scheduling = 2.14	E B
3rd PM (Process A, Set 2) = 3.13	3rd Scheduling = 1.34	E B A
4th PM (Process C, Set 2) = 2.57	4th Scheduling = 1.87	E B A C
5th PM (Process G, Set 2) = 2.45	5th Scheduling = 1.21	E B A C G
6th PM (Process D, Set 1) = 2.33	6th Scheduling = 2.18	E B A C G D
7th PM (Process F, Set 1) = 2.02	7st Scheduling = 4.15	E B A C G D F

Figure 1. Example of the AutoMig's approach

The main part of AutoMig concerns its prediction function *pf*. *pf* emulates the time of a superstep by analyzing the computation and communication parts of the processes. Both parts are computed through Equations 5 and 6, respectively. They work with data collected at the superstep before calling the rescheduling facility. In addition, *pf* considers information about the migration costs of the processes to the

Sets. The final selection of migratable processes is obtained through verifying the lowest *pf*. The processes in the level belonging to this prediction are elected for migration if their rescheduling outperforms the *pf* for the current mapping.

At the rescheduling call, each process passes the following data to its manager: (i) its highest *PM*; (ii) a vector with its migration costs (Mem metric) for each Set; (iii) the number of instructions; (iv) a vector which contains the number of bytes involved on communication actions to each Set. Each manager exchanges *PM* values and uses them to create a decreasing-sorted list. Task 5 of Algorithm 1 is responsible for getting data to evaluate the current scheduling.

At each level of the *PM* list, the data of the target process is transferred to the destination Set. For instance, data from process 'E' is transferred to Set 2 according to the example illustrated in Figure 1. Thus, the manager on the destination Set will choose a suitable processor for the process and will calculate Equations 5 and 6 for it. Aiming to minimize multicast communication among the managers at each *pf* computation, each Set Manager computes $Time_p$ and $Comm_p$ for the processes under its jurisdiction and save the results together with the specific level of the list. After performing the tasks for each element on *PM* list, the managers exchange vectors and compute *pf* for each level as well as for the present scheduling (task 12 in Algorithm 1).

Equation 5 computes $Time_p(i)$, where i means a specific process. $Time_p(i)$ uses data related to the computing power and the load of the processor in which process i executes currently or is being tested for rescheduling. $cpu\_load(i)$ represents the CPU load average on the last 15 minutes. This time interval was adopted based on work of Vozmediano and Conde [9]. Equation 6 presents how we get the maximum communication time when considering process i and Set j. In this context, Set j may be the current Set of process i or a Set in which this process is being evaluated for migration. $T(k, j)$ refers to the transferring rate of 1 byte from the Set Manager of Set j to other Set Manager. $Bytes(i, k)$ works with the number of bytes transferred through the network among process i and all process belonging to Set k. Lastly, $Mig\_Costs(i, j)$ denotes the migration costs related to the sending of process i to Set j. It receives the value of the Mem metric, which also considers a process i and a Set j.

$$Time_p(i) = \frac{Instruction(i)}{(1 - cpu\_load(i)).cpu(i)} \qquad (5)$$

$$\begin{aligned} Comm_p(i,j) \quad &= \quad Max_k \ (\ \forall \ k \in Sets \\ & \qquad (Bytes(i,k) \ . \ T(k,j)) \ ) \end{aligned} \qquad (6)$$

$$\begin{aligned} pf \quad &= \quad Max_i \ (Time_p(i)) + Max_{i,j} \ (Comm_p(i,j)) \\ &+ \quad Max_{i,j} \ (Mig\_Costs(i,j)) \end{aligned} \qquad (7)$$

Considering Equation 7, we can emphasize that each part may consider a different process i and Set j. For instance, a specific process may obtain the largest computation time, while other one expends more time in communication actions. AutoMig uses a global strategy, where data from all processes are considered in the calculus. We take profit from the barriers of the BSP model for exchanging scheduling data, not paying additional costs for that.

Algorithm 1 AutoMig's approach for selecting the processes

```
1: Each process computes PM locally (see Equation 4).
2: Each process passes its highest PM, together with the
   number of instructions and a vector that describes its
   communication actions, to the Set Manager.
3: Set Managers exchange PM data of their processes.
4: Set Managers create a sorted list based on the PM values
   with $n$ elements ($n$ is the number of processes).
5: Set Managers compute Eq. 5 and 6 for their processes.
   The results will be used later for measuring the current
   mapping. Migrations costs are not considered.
6: for each element from 0 up to $n - 1$ in the PM list do
7:    Considered element is analyzed. Set Manager of
      process $i$ sends data about it to the Set Manager of Set
      $j$. The algorithm proceeds its calculus by considering
      that process $i$ is passed to Set $j$.
8:    The manager on the destination Set chooses a suitable
      processor to receive the candidate process $i$.
9:    Set Managers compute Eq. 5 and 6 for their processes.
10:   Set Managers save the results in a vector with the
      specific level of the PM list.
11: end for
12: Set Managers exchange data and compute pf for the
    current scheduling as well as for each level on PM list.
13: if $Min(pf)$ in the PM list $<$ current pf then
14:    Considering the PM list, the processes in the level
       where pf was reached are selected for migration.
15:    Managers notify their elected processes to migrate.
16: else
17:    Migrations do not take place.
18: end if
```

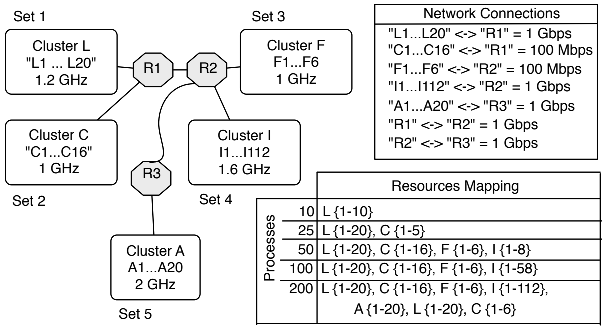
IV. Evaluation Methodology

We are simulating the functioning of a BSP-based Fractal Image Compression (FIC) application [7]. FIC applications apply transformations which approximate smaller parts of the image by larger ones. The smaller parts are called ranges and the larger ones domains. All ranges together form the image. The domains can be selected freely within the image. A complete domain-poll of an image of size $t \times t$ with square domains of size $d \times d$ consists of $(t - d + 1)^2$ domains. Furthermore, each domain has 8 isometries. So each range is compared with $8(t-d+1)^2$ domains. The application time increases as the number of domains increases as well. Our BSP modeling considers the variation of both the range and domain sizes as well as the number of processes. Algorithm

2 presents the organization of a single superstep. Firstly, we are computing $\frac{t}{r}$ supersteps, where $t \times t$ is the image size and r is the size of square ranges. The goal is to compute a set of ranges at each superstep. For that, each superstep works over $\frac{t}{r}$ ranges since the image comprises a square. At each superstep, a range is computed against $8((\frac{t}{d})^2.\frac{1}{n})$ domains, where d represents the size of a domain and n the number of processes. Thus, each process sends $\frac{t}{r}$ ranges before calling the barrier, which must be multiplied by 8 to find the number of bytes (each range occupies 8 bytes).

Algorithm 2 Single superstep for FIC problem

```
Taking a range-pool $rp$ ($0 \leq rp \leq \frac{t}{r} - 1$): $t$ and $r$ mean
  the sides of the $t \times t$ image and $r \times r$ range, respectively
for each range in $rp$ do
  for each domain belonging to specific process do
    for each isometry of a domain do
      calculate-rms(range, domain)
    end for
  end for
end for
Each process $i$ ($0 \leq i \leq n - 1$) sends data to its right-
  neighbor $i + 1$. Process $n - 1$ sends data to process $0$
Call for synchronization barrier
```



Network Connections

"L1...L20" <-> "R1" = 1 Gbps
"C1...C16" <-> "R1" = 100 Mbps
"F1...F6" <-> "R2" = 100 Mbps
"I1...I112" <-> "R2" = 1 Gbps
"A1...A20" <-> "R3" = 1 Gbps
"R1" <-> "R2" = 1 Gbps
"R2" <-> "R3" = 1 Gbps

Processes	Resources Mapping
10	L {1-10}
25	L {1-20}, C {1-5}
50	L {1-20}, C {1-16}, F {1-6}, I {1-8}
100	L {1-20}, C {1-16}, F {1-6}, I {1-58}
200	L {1-20}, C {1-16}, F {1-6}, I {1-112}, A {1-20}, L {1-20}, C {1-6}

Figure 2. Multiple Clusters-based topology, processing and network resources description and the initial processes-resources scheduling

The BSP application was evaluated with simulation in three scenarios: (i) Application execution simply; (ii) Application execution with MigBSP scheduler without applying migrations; (iii) Application execution with MigBSP scheduler allowing migrations. Both the application and AutoMig were developed using the SimGrid Simulator (MSG Module) [3]. It is deterministic, where a specific input always results in the same output. The scenarios were evaluated in an infrastructure with five Sets (see Figure 2). A Set represents a cluster where each node has a single processor.

The infrastructure permits us to analyze the impact of the heterogeneity issue on AutoMig's algorithms.

Initial tests were executed using α equal to 4 and D equal to 0.5. We observed the behavior of 10, 25, 50, 100 and 200 BSP processes. Their initial mapping to the resources may be viewed in Figure 2. Since the application proceeds in communications from process i to $i + 1$, we are using the contiguous approach in which a cluster is filled before passing to another one [10]. The values of 40, 20 and 10 were used for the side (d) of a square domain and the figure is a square 1000x1000. The lower the d value, the greater the number of domains for computation. Finally, the migration costs are based on tests with AMPI in our clusters.

V. Analyzing AutoMig's Overhead and Decisions

Table I presents the tests with 40 and 20 for both domain and range sizes, respectively. This setup enables a small computation grain and processes migrations are not viable. *PM* values in all situations are negative, owing to the lower weight of the computation and communication actions if compared to the migration costs. AutoMig figures out the lowest *pf* for the current scheduling. Thus, both times for scenario ii and iii are higher than scenario i. In this context, a large overhead is imposed by MigBSP since the normal application execution is close to 1 second in average.

Table I
Results with 40 for domain (time in seconds)

Processes	Scenario i	Scenario ii	Scenario iii
10	1.20	2.17	2.17
25	0.66	1.96	1.96
50	0.57	2.06	2.06
100	0.93	2.44	2.44
200	1.74	3.41	3.41

We increase the number of domains when dealing with 20 for the domain's side. The execution with 20 for domain is depicted in Figure 3. The execution with 10 processes did not present replacement because they are balanced. *pf* of 0.21 was obtained for the current processes-resources mapping by using 20 for domain and 10 processes. All predictions in the *PM* list are higher than 0.21 and their average achieves 0.38. However, this configuration of domain triggers migration when using 25 and 50 processes. In the former case, 5 processes from cluster C are moved to the fastest cluster named A. AutoMig's decisions led a gain of 17.15% with process rescheduling in this context. The last mentioned cluster receives all processes from cluster F when dealing with 50 processes. This situation shows up gains of 12.05% with migrations. All processes from cluster C remain on their initial location because the computation grain decreases with 50 processes. Although 14 nodes in the fastest cluster A stay free, AutoMig does not select some processes for execution on them because the BSP model presents a barrier. Despite 14 migrations from cluster C to A occur, a group of process in the slower cluster will remain inside it and still limit the superstep's time. Lastly, since the work grain decreases when adding more processes, the executions with 100 and 200 did not present migrations.

We achieved better results when using 10 for domain (see Figure 4). The computation grain increases exponentially with this configuration. This sentence may be viewed through the execution of 10 processes, in which are all migrated to cluster A. Considering that $8((\frac{t}{d})^2.\frac{1}{10})$ express the number of domains assigned to each one of 10 processes, this expression is equal to 500, 2000 and 8000 when testing 40, 20 and 10 values for domain. Using 10 for both domain and the number of processes, the current scheduling produced a *pf* of 1.62. *pf* for the *PM* list is shown below:

- $pf[1..10] = \{$1.79, 1.75, 1.78, 1.79, 1.81, 1.76, 1.74, 1.82, 1.78, 1.47$\}$.

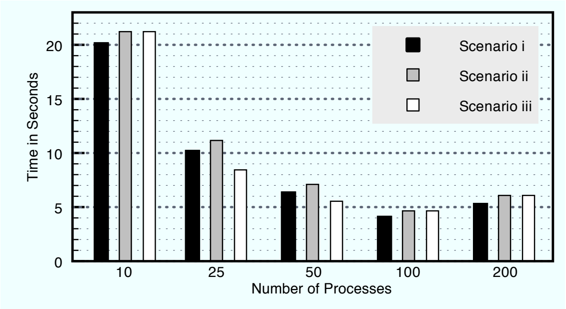


Figure 3. AutoMig's evaluation when using 20 for domain

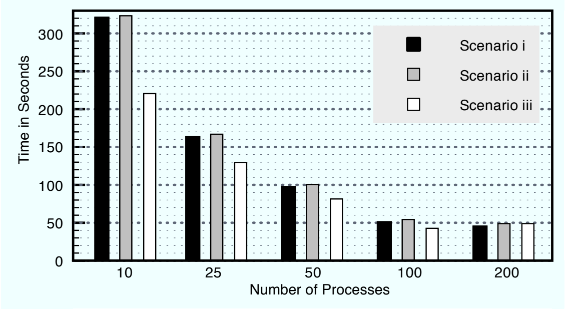


Figure 4. AutoMig's results when enlarging the work per process at each superstep. This graph illustrates experiments with domain 10 and range 5

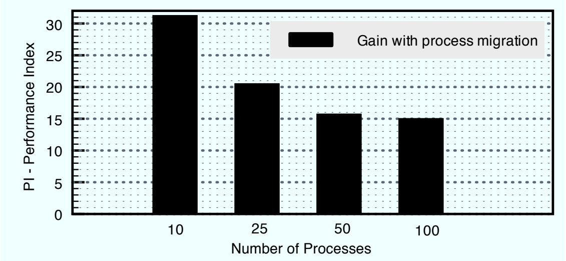


Figure 5. Migration gains with domain 10. PI =$(\frac{scen.\ i - scen.\ iii}{scen.\ i} * 100)$

Considering the first up to the ninth *pf*in the last itemization, we observed that although some processes can run faster in a more appropriate cluster, there are others that remain in a slower cluster. This last group does not allow performance gains due to the BSP modeling. However, the migration of 10 processes to the fastest cluster generates a *pf* of 1.47 and a gain around 31.13% when comparing scenarios iii and i. This analysis is illustrated in Figure 5. The processes from cluster C are moved to A with 25 processes and domain equal to 10. In this case, the 20 other processes stay on cluster L because there are not enough free nodes in the fastest cluster. A possibility is to explore two process in a node of cluster A (each node has 2 GHz) but AutoMig does not apply it because each node in Cluster L has 1.2 GHz. Considering the growth in the number of domains, the migrations with 100 processes becomes viable and get 14.95% of profit. Nevertheless, the initial mapping of 200 processes stands the same and an overhead of 7.64% was observed when comparing both scenarios i and ii.

We can conclude that the higher the computation weight per process, the better will be the gains with process rescheduling. In this way, we tested AutoMig with a shorter domain as expressed in Table II. This table shows the behavior for 10 and 25 processes. Gains about 31.62% and 19.81% were obtained when dealing with AutoMig. In addition, its overhead is shorter than 1%. We verified that the benefits with migrations remain practically constant if we compare the executions with 10 and 4 for the domain values. It is possible to observe that when doubling the number of processes, the application time is not halved as well.

Table II
EXECUTION TIME (IN SECONDS) WITH DOMAIN 4

Proc-esses	Scen.i	Old Heuristic		AutoMig	
		Scen.ii	Scen.iii	Scen.ii	Scen.iii
10	12500.51	12511.87	9191.72	12523.22	8555.29
25	6250.49	6257.18	5311.54	6265.38	5011.77

Table II also shows a comparative analysis of the two selection heuristics implemented in MigBSP. We named the one that selects one process at each rescheduling call as Old Heuristic. Despite both obtained good levels of performance, AutoMig achieves better migration results than Old Heuristic (approximately 8%). For instance, 5 processes are migrated already in the first attempt for migration when testing 25 processes. In this case, all processes that were running on Cluster C are passed to Cluster A. This reorganization suggested by AutoMig at the beginning of the application provides a shorter time for application conclusion. In the other hand, 5 rescheduling calls are needed to reach the same configuration expressed previously with Old Heuristic. Lastly, AutoMig imposes larger overheads if compared to Old Heuristic (close to 1%). This situation was expected since two multicast communications among the Set Managers are performed by AutoMig in its algorithms.

VI. RELATED WORK

Vadhiyar and Dongarra presented a migration framework and self-adaptivity in GrADS system [12]. The gain with rescheduling is based on the remaining execution time prediction over a new specified resource. This framework must work with applications in which their parts and durations are known in advance. Sanjay and Vadhiyar [11] present a scheduling algorithm called Box Elimination. It considers a 3-D box of CPU, bandwidth and processors tuples for selecting the resources with minimum available CPU and bandwidth. This work treats applications in which the problem size is known in advance. Liu et al. [8] introduced a novel algorithm for resource selection. The application reports the Execution Satisfaction Degree (ESD) to the scheduling middleware. The main weakness of this idea is the fact that users/developers need to define the ESD function by themselves for each new application.

Concerning the migration context, PUBWCL [2] and PUB [1] libraries enable this facility on BSP applications. PUBWCL aims to take profit of idle cycles from nodes around the Internet [2]. All proposed algorithms just use data about the computation times of each process as well as data regarding the nodes's load. The PUB's author proposed both centralized and distributed strategies for load balancing. Both strategies consider neither the communication among the processes, nor the migration costs.

VII. Conclusion and Future Work

Considering that the bulk synchronous style is a common organization for MPI programs [1], [6], AutoMig emerges as an alternative for selecting their processes for running on more suitable resources without interferences from the developers. AutoMig's main contribution appears on its prediction function *pf*. *pf* is applied for the current scheduling as well as for each level of a Potential of Migration-based list. Each element of this list informs a new scheduling through the increment of one process replacement. *pf* considers the load on both the Sets and the network, estimates the slowest processes regarding their computation and communication actions and adds the migration costs. The key problem to solve may be summarized in maintaining the current processes' location or to choose a level of the list.

AutoMig's load balancing scheme uses the global approach, where data from all processes are considered in the calculus [13]. Instead to pay a synchronization cost to get the scheduling information, AutoMig takes profit from the BSP superstep concept in which a barrier always occurs after communication actions. AutoMig and an application were developed using the SimGrid Simulator. Since the application is CPU-bound, the shorter the domain's size the higher the application's time and migration profitability. The results proved this, indicating gains up to 17.15% and 31.13% for domains equal 20 and 10. Particularly, the results revealed the main AutoMig's strength on selecting the migratable processes. It can elect the whole set of processes belonging to a slower cluster to run faster in a more appropriate one. But, sometimes a faster cluster has fewer free nodes than the number of candidates. Migrations do not take place in this situation owing to the execution rules of a BSP superstep.

Finally, future work comprises the use of AutoMig in a HPC service for Cloud computing. Concerning that each application specifies its own SLA previously, AutoMig appears as the first initiative to reorganize the processes-resources shaping when SLA fails. If the rescheduling does not solve the problem, more resources are allocated in a second instance.

VIII. Acknowledgments

This work is partially supported by CTIC RNP and CNPq.

References

[1] O. Bonorden. Load balancing in the bulk-synchronous-parallel setting using process migrations. In *21th International Parallel and Distributed Processing Symposium (IPDPS 2007)*, pages 1–9. IEEE, 2007.

[2] O. Bonorden, J. Gehweiler, and F. M. auf der Heide. Load balancing strategies in a web computing environment. In *Proceeedings of International Conference on Parallel Processing and Applied Mathematics (PPAM)*, pages 839–846, Poznan, Poland, September 2005.

[3] H. Casanova, A. Legrand, and M. Quinson. Simgrid: A generic framework for large-scale distributed experiments. In *Int. Conf. on Computer Modeling and Simulation (uksim)*, pages 126–131, 2008. IEEE.

[4] R. da Rosa Righi, L. L. Pilla, A. Carissimi, P. A. Navaux, and H.-U. Heiss. Observing the impact of multiple metrics and runtime adaptations on bsp process rescheduling. *Parallel Processing Letters*, 20(2):123–144, June 2010.

[5] F. J. da Silva, F. Kon, A. Goldman, M. Finger, R. Y. de Camargo, F. C. Filho, and F. M. Costa. Application execution management on the integrade opportunistic grid middleware. *J. Parallel Distrib. Comput.*, 70(5):573–583, 2010.

[6] R. E. De Grande and A. Boukerche. Dynamic balancing of communication and computation load for hla-based simulations on large-scale distributed systems. *J. Parallel Distrib. Comput.*, 71:40–52, 2011.

[7] Y. Guo, X. Chen, M. Deng, Z. Wang, W. Lv, C. Xu, and T. Wang. The fractal compression coding in mobile video monitoring system. In *CMC '09: Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, pages 492–495, Washington, DC, USA, 2009. IEEE Computer Society.

[8] H. Liu, S.-A. Sørensen, and A. Nazir. On-line automatic resource selection in distributed computing. In *Int. Conf. on Cluster Computing*, pages 1–9. IEEE, 2009.

[9] R. Moreno-Vozmediano and A. B. Alonso-Conde. Influence of grid economic factors on scheduling and migration. In *High Perf. Comp. for Computational Science - VECPAR*, pages 274–287. Springer, 2005.

[10] J. A. Pascual, J. Navaridas, and J. Miguel-Alonso. Job scheduling strategies for parallel processing. chapter Effects of Topology-Aware Allocation Policies on Scheduling Performance, pages 138–156. Springer, 2009.

[11] H. A. Sanjay and S. S. Vadhiyar. A strategy for scheduling tightly coupled parallel applications on clusters. *Concurr. Comput. : Pract. Exper.*, 21(18):2491–2517, 2009.

[12] S. S. Vadhiyar and J. J. Dongarra. Self adaptivity in grid computing: Research articles. *Concurr. Comp. : Pract.Exper.*, 17(2-4):235–257, 2005.

[13] M. J. Zaki, W. Li, and S. Parthasarathy. Customized dynamic load balancing for a network of workstations. *J. Parallel Distrib. Comput.*, 43(2):156–162, 1997.

ISBN: 978-1-61208-161-8

Min-Sum-Min Message-Passing for Quadratic Optimization

Guoqiang Zhang and Richard Heusdens
Department of Mediamatics
Delft University of Technology
Delft, the Netherlands
Email: {g.zhang-1,r.heusdens}@tudelft.nl

***Abstract*—We study the minimization of a quadratic objective function in a distributed fashion. It is known that the min-sum algorithm can be applied to solve the minimization problem if the algorithm converges. We propose a min-sum-min message-passing algorithm which includes the min-sum algorithm as a special case. As the name suggests, the new algorithm involves two minimizations in each iteration as compared to the min-sum algorithm which has one minimization. The algorithm is derived based on a new closed-loop quadratic optimization problem which has the same optimal solution as the original one. Experiments demonstrate that our algorithm improves the convergence speed of the min-sum algorithm by properly selecting a parameter in the algorithm. Furthermore, we find empirically that in some situations where the min-sum algorithm fails, our algorithm still converges to the right solution. Experiments show that if our algorithm converges, our algorithm outperform a reference method with fast convergence speed.**

***Keywords*-Distributed optimization, Gaussian belief propagation, message-passing algorithms**

I. Introduction

In this paper we consider minimizing a quadratic optimization problem, namely

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^T Jx - h^T x, \quad (1)$$

where $J \in \mathbb{R}^{n\times n}$ is a positive definite matrix and $h \in \mathbb{R}^n$. It is known that the optimal solution x^* satisfies a linear equation

$$Jx^* = h.$$

We suppose that the matrix J is sparse and the dimensionality n is large. In this situation, the direct computation (without using the sparse structure of J) of the optimal solution may be expensive and unscalable. One natural question is how to exploit the sparse geometry to efficiently obtain the optimal solution. To achieve this goal, the quadratic function $f(x)$ can be associated with an undirected graph $G = (V, E)$. That is, the graph has a node for each variable x_i and an edge between i and j for each nonzero J_{ij} term. The algorithms that exploit the sparse geometry exchange information between nodes in the graph until reaching consensus.

Existing algorithms are either applicable to a specific class of J or are computationally expensive (which we will explain in detail in next section). Our work will focus on designing an efficient distributed message-passing algorithm for a general positive definite matrix J.

The reminder of the paper is organized as follows. In Section II, we provide a literature review. Section III briefly describes the GaBP algorithm, or equivalently, the min-sum algorithm for quadratic optimization. In Section IV, we present our new min-sum-min message-passing algorithm. Section V provides the experimental results. Finally, we draw conclusions in Section VI.

II. Related Work

The quadratic optimization problem is closely related to the Gaussian belief propagation (GaBP) for inference in graphic models. This is due to the fact that $f(x)$ can be associated with a Gaussian distribution $p(x)$ via

$$p(x) \propto \exp(-(1/2)x^T Jx + h^T x).$$

The mean value of $p(x)$ is the same as the optimal solution of the quadratic optimization problem. The GaBP algorithm is a min-sum message-passing algorithm for estimating the mean of the Gaussian random vector. Due to its simplicity, the GaBP algorithm has found many applications in practice, such as signal processing [1][2], consensus propagation in sensor networks [3], multiuser detection [4] and Turbo decoding with Gaussian densities [5]. It is known that if the GaBP algorithm converges, it converges to the mean value of $p(x)$ (see [6],[7]). Unfortunately, the GaBP algorithm does not always converge, which limits its application. Two general sufficient conditions for the convergence of the GaBP algorithm are established: diagonal dominance of J [8] and walk-summability of J [9][6]. For completeness, we give their definitions in the following.

Definition [8],[10] A matrix $J \in \mathbb{R}^{n\times n}$, with all ones on its diagonal, is walk-summable if the spectral radius of the matrix $\bar{J} - I$, where $\bar{J} = [|J_{ij}|]_{i,j=1}^n$, is less than one.

Definition [10] A matrix $J \in \mathbb{R}^{n\times n}$ is diagonally dominant if $|J_{ii}| > \sum_{j\neq i} |J_{ij}|$ for all i.

Recently research attention has moved to overcome the convergence-failure of the GaBP algorithm for a general matrix J. In [10], Ruozzi and Tatikonda proposed a variant

ISBN: 978-1-61208-161-8

of the GaBP algorithm by changing the edge structure of the graph. In their algorithm, two parameters have been introduced to ensuring the correct convergence. However, it is not clear how to choose the two parameters. In [11], Johnson et al. proposed a double-loop algorithm with the GaBP algorithm as a subroutine (corresponds to the inner loop). Each time the GaBP algorithm is called a better estimate of the mean vector is obtained. The double-loop algorithm guarantees the convergence at the cost of high computational complexity. The basic idea of the double-loop algorithm is to precondition the matrix J such that the new matrix is diagonal dominant, allowing the use of the GaBP algorithm.

In this paper we generalize the min-sum algorithm by proposing a new min-sum-min algorithm. We first construct a new closed-loop quadratic optimization problem which has the same optimal solution as that of the original problem. Instead of solving the original problem, we solve the new problem by developing the min-sum-min algorithm. The basic idea behind the algorithm is to transform the closed-loop optimization problem into n scalar closed-loop optimization problems, one for each node. Note that our algorithm has two minimizations for each iteration as compared to the min-sum algorithm which has one minimization. The additional minimization in our algorithm serves to break the loop at each node.

We test our algorithm for two scenarios. When the min-sum algorithm converges, we find that our algorithm can be more efficient than the min-algorithm by properly choosing a parameter in the algorithm. When the min-sum algorithm fails, we find that our algorithm still converges in some situations. Experiments show that our algorithm significantly improves the convergence speed of the double-loop algorithm [11].

III. Min-Sum Message-Passing

In this section we briefly review the min-sum message-passing algorithm for quadratic optimization (which is actually the GaBP algorithm). This algorithm is the basis for developing our new min-sum-min message-passing algorithm.

Before considering the quadratic optimization problem (1), we first study a more general objective function $f(x)$, which takes the form

$$f(x) = \sum_{j\in V} f_j(x_j) + \sum_{(i,j)\in E} f_{ij}(x_i,x_j). \qquad (2)$$

In the literature, f_j and f_{ij} are often called self-potentials and edge potentials, respectively. f_{ij} captures the correlation between nodes i and j. Due to the pairwise correlations, the jth component x_j^* of x^* that minimizes $f(x)$ requires a global knowledge of $f(x)$. The min-sum algorithm describes the form of the messages exchanged between the nodes. Specifically, the message sent from node i to node j at iteration $t+1$ takes the form

$$\begin{aligned} m_{i\to j}^{(t+1)}(x_j) \;=\; \kappa + \min_{x_i}\Bigg(& f_i(x_i) \\ & + f_{ij}(x_i,x_j) + \sum_{u\in N(i)\backslash j} m_{u\to i}^{(t)}(x_i)\Bigg), \end{aligned} \qquad (3)$$

where $N(i)$ denotes the set of neighboring nodes of i, i.e., $N(i) = \{j|(i,j)\in E\}$. The parameter κ in (3) represents an arbitrary offset term that may be different from message to message. (3) implies that there are two messages associated with each edge $(i,j)\in E$, one for each direction on the edge. To facilitate the performance analysis, we introduce a directed graph $\vec{G} = (V,\vec{E})$ for G. For every edge $(i,j)\in E$, there are two elements $(i\to j)$, $(j\to i)$ in $\vec{E}$.

At each time t, each vertex j forms a local belief function $f_j^{(t)}(x_j)$ by combining messages received from all neighbors

$$f_j^{(t)}(x_j) = f_j(x_j) + \sum_{u\in N(j)} m_{u\to j}^{(t)}(x_j). \qquad (4)$$

An estimate of the jth component x_j^* is then given by

$$\hat{x}_j^{(t)} = \arg\min_{x_j} f_j^{(t)}(x_j). \qquad (5)$$

The min-sum algorithm is successful if $\hat{x}_j^{(\infty)}$ is equal to x_j^* for all $j\in V$.

When the function $f(x)$ in (2) is specified to the quadratic function as in (1), the min-sum algorithm becomes the GaBP algorithm. In this situation, the self-potentials and edge potentials are given by

$$\begin{aligned} f_j(x_j) &= (1/2)J_{jj}x_j^2 - h_j x_j \\ f_{ij}(x_i,x_j) &= J_{ij}x_i x_j. \end{aligned}$$

Without loss of generality, we may assume that J is normalized to have unit diagonal, i.e., $J_{jj}=1$. Since the functions f_j and f_{ij} are in quadratic form, the belief function $f_j^{(t)}$ also takes a quadratic form [7]:

$$\begin{aligned} f_j^{(t)}(x_j) \quad &= \quad \frac{1}{2}\left(1 - \sum_{i\in N(j)} J_{ij}^2\gamma_{ij}^{(t)}\right)x_j^2 \\ &\qquad - \left(h_j - \sum_{i\in N(j)} z_{ij}^{(t)}\right)x_j, \end{aligned} \qquad (6)$$

where $\gamma_{ij}^{(t)}$ and $z_{ij}^{(t)}$ are updated as

$$\gamma_{ij}^{(t+1)} = \frac{1}{1-\sum_{u\in N(i)\backslash j} J_{ui}^2\gamma_{ui}^{(t)}}, \qquad (7)$$

$$z_{ij}^{(t+1)} = \frac{J_{ij}}{1-\sum_{u\in N(i)\backslash j} J_{ui}^2\gamma_{ui}^{(t)}}\left(h_i - \sum_{u\in N(i)\backslash j} z_{ui}^{(t)}\right). \qquad (8)$$

The update of $\gamma_{ij}^{(t+1)}$ and $z_{ij}^{(t+1)}$ are valid if $\sum_{u\in N(i)\backslash j} J_{ui}^2\gamma_{ui}^{(t)} < 1$ for all i, j and t. These inequalities are always satisfied under the walk-summability condition or the diagonal dominant condition [8],[10]. Given the form of the belief function $f_j^{(t)}(x_j)$ in (6), the estimate of x_j^* is obtained by applying (5)

$$\hat{x}_j^{(t)} = \frac{1}{1-\sum_{i\in N} J_{ij}^2\gamma_{ij}^{(t)}}\left(h_j - \sum_{i\in N(j)} z_{ij}^{(t)}\right). \quad (9)$$

The message-updating equations (6)-(9) are described above for comparison with our algorithm in Section IV. We will explain how our min-sum-min algorithm is derived based on the min-sum messages (3)-(5).

IV. Min-Sum-Min Message-Passing

In this section we first construct a new closed-loop quadratic minimization problem. The new problem has the same optimal solution as that of the original problem. We then propose a so-called min-sum-min message-passing algorithm for the new problem. Finally, we provide explicit message-updating expressions for solving the new problem.

A. A New Cost Function

Based on (1), we define a new quadratic minimization problem:

$$x^* = \arg\min_x \tilde{f}(x,x^*), \quad (10)$$

where

$$\tilde{f}(x,x^*) = \frac{1}{2}x^T(sI+(1-s)J)x-[(1-s)h+sx^*]^T x, \quad (11)$$

where s is a scalar parameter and I is the identity matrix. Different from (1), the optimal solution x^* appears on both sides of (10). Thus, (10) is in fact a closed-loop optimization problem. It is obvious that the min-sum algorithm cannot be directly applied here since x^* is not known. We explain in the following how $\tilde{f}(x,x^*)$ is constructed as in (11).

Before providing the motivation for $\tilde{f}(x,x^*)$, we first show that the optimal solution of (10) is the same as that of the original minimization problem. We let $\tilde{J}_s = sI+(1-s)J$. Same as J, the new matrix $\tilde{J}_s$ also has unit diagonal. In order that the new optimization problem is well defined, we choose s such that $\tilde{J}_s$ is positive definite. It should be noted that s can be negative depending on J. To solve (10), we first fix x^* in $\tilde{f}(x,x^*)$. We then set the first derivative of $\tilde{f}(x,x^*)$ w.r.t. x to be 0. By doing so, we have

$$\begin{aligned}[sI+(1-s)J]\,x^* &= (1-s)h+sx^*,\\ Jx^* &= h.\end{aligned}$$

Thus instead of solving (1), we can solve the new optimization problem.

Note that the introduction of sx^* in constructing $\tilde{f}(x,x^*)$ is the key point in designing a new message-passing algorithm. Due to the simple form of sx^*, the self-potentials and edge potentials of $\tilde{f}(x,x^*)$ also take a simple form:

$$\begin{aligned}\tilde{f}_j(x_j,x_j^*) &= (1/2)x_j^2-[(1-s)h_j+sx_j^*]x_j, \quad (12)\\ \tilde{f}_{ij}(x_i,x_j) &= (1-s)J_{ij}x_ix_j. \quad (13)\end{aligned}$$

We point out that the self-potential $\tilde{f}_j(x_j,x_j^*)$ has only x_j^* involved instead of the whole vector x^*. This property of $\tilde{f}_j(x_j,x_j^*)$ makes it possible for node j to deal with x_j^* locally. In fact, one can introduce Γx^*, where Γ is a diagonal matrix, in constructing a new closed-loop function (the matrix $\tilde{J}_s$ should be changed accordingly). For the same reason, one can also design a massage-passing algorithm. In this paper we focus on sx^* for simplicity.

We point out that the diagonal-loading on J to obtain $\tilde{J}_s$ is inspired by the work in [11]. The main difference between our work and [11] is that we propose a new message-passing algorithm based on (10). On the other hand, the authors in [11] took the min-sum algorithm as a subroutine to solve (1) directly.

B. Algorithm design

In this subsection we present the min-sum-min algorithm for solving the closed-loop optimization problem (10). We show that one of the two minimizations of the algorithm unlocks the loopy effect of x^* in (10).

In order to tackle the unknown parameters x_j^* in self-potentials $\tilde{f}_j(x_j,x_j^*)$, we first revisit the min-sum algorithm as described by (3)-(5). Note that after each iteration of message-passing, an estimate $\hat{x}_j^{(t)}$ of x_j^* can be obtained from the local belief function $f_j^{(t)}(x_j)$. In other words, a new estimate of x_j^* is always accessible to node j after each iteration. Inspired by this property of the min-sum algorithm, we propose to compute an estimate of x_j^* in (12) at each iteration in designing our new algorithm. We then take the estimate of x_j^* for message-updating in the next iteration. In principle, if the estimate of x_j^* becomes more and more accurate as the information diffuses through message-passing, the algorithm converges to the right solution.

Based on the above analysis, we propose new message-updating expressions as

$$\hat{x}_i^{(t)} = \arg\min_{x_i}\left(\tilde{f}_i(x_i,\hat{x}_i^{(t)})+\sum_{u\in N(i)}\tilde{m}_{u\to i}^{(t)}(x_i)\right), (14)$$

$$\check{x}_i^{(t)} = g_i\left(\hat{x}_i^{(t)},\hat{x}_u^{(t)},u\in N(i)\right), \quad (15)$$

$$\begin{aligned}\tilde{m}_{i\to j}^{(t+1)}(x_j) = \kappa + \min_{x_i}\Bigg(&\tilde{f}_i(x_i,\check{x}_i^{(t)})\\ &+\tilde{f}_{ij}(x_i,x_j)+\sum_{u\in N(i)\backslash j}\tilde{m}_{u\to i}^{(t)}(x_i)\Bigg). \quad (16)\end{aligned}$$

Note that there are two minimization operations in (14)-(16) for each iteration, as compared to (3) which has only one minimization. The name *min-sum-min* for our new algorithm then arises naturally. The first minimization in (14) comes from (5) and (12). This minimization plays an important role in breaking the loop in (10). The second minimization in (16) comes from the min-sum algorithm. The function g_i in (15) is utilized to refine the estimate using the outputs of the first minimization. (14)-(15) together provide an estimate of x_i^* for node i at each iteration.

Note that (14) is again a closed-loop minimization with respect to $\hat{x}_i^{(t)}$. Thus we successfully transform the global closed-loop optimization problem into n local closed-loop optimization problems, one for each node. As all the messages are in quadratic form, it is not difficult to compute $\hat{x}_i^{(t)}$ after each iteration. Once $\{g_i | i \in V\}$ is specified, we effectively provide a min-sum-min algorithm to solve (10).

We can also interpret (14)-(16) from another viewpoint. Note that (14)-(15) combines information from neighboring nodes in computing an estimate of the optimal solution. Thus (14)–(15) can be viewed as an information-fusion step. On the other hand, the second minimization (16) carries information from node i to a neighboring node j. Correspondingly, (16) can be viewed as an information-diffusion step. The two steps are implemented in order until reaching consensus at each individual node. That is, the estimate $\check{x}_i^{(t)}$ of the optimal component x_i^* is stable over time for all i.

Remark 1: The min-sum-min algorithm is a natural extension of the min-sum algorithm. To see this, we let s approach to 0, it is immediate that $\tilde{m}_{ij}(x_j) \to m_{ij}(x_j)$. Since our algorithm has a free parameter s to choose, we can improve the performance of the algorithm by properly adjusting the parameter.

Remark 2: Note that the min-sum-min algorithm is not limited to the quadratic minimization problem. In fact, as long as an original optimization problem can be reformulated into a proper closed-loop optimization problem, the min-sum-min algorithm can be applied in correspondence.

C. *Explicit message-updating expressions*

In this subsection we provide explicit message-updating expressions for solving the closed-loop optimization problem. We study the three updating expressions (14)-(16) one by one for the quadratic form of the potentials (12)-(13).

We first consider the minimization (14). We suppose that the message $\tilde{m}_{u\to i}^{(t)}(x_i)$ at iteration t takes the form

$$\tilde{m}_{u\to i}^{(t)}(x_i) = -\frac{1}{2}(1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)} x_i^2 + \tilde{z}_{ui}^{(t)} x_i, \qquad (17)$$

where $\tilde{\gamma}_{ui}^{(t)}$ and $\tilde{z}_{ui}^{(t)}$ are the associated parameters characterizing the quadratic form. By plugging (17) into (14), we obtain

$$\begin{aligned}\hat{x}_i^{(t)} \;=\; & \arg\min_{x_i} \Bigg(\frac{1}{2}\Big[1 - \sum_{u\in N(i)} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}\Big] x_i^2 \\ & - \Big[(1-s)h_i + s\hat{x}_i^{(t)} - \sum_{u\in N(i)} \tilde{z}_{ui}^{(t)}\Big] x_i \Bigg). \qquad (18)\end{aligned}$$

The optimal solution $\hat{x}_i^{(t)}$ can be easily computed from (18), expressed as

$$\hat{x}_i^{(t)} = \frac{(1-s)h_i - \sum_{u\in N(i)} \tilde{z}_{ui}^{(t)}}{1 - s - \sum_{u\in N(i)} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}}. \qquad (19)$$

Given the expression for $\hat{x}_i^{(t)}$, we then specify the function set $\{g_i | i \in V\}$ in (15). To achieve this goal, we construct an equality

$$x^* = \frac{1}{2}\Big(h + x^* - (J - I)x^*\Big), \qquad (20)$$

where x^* is the optimal solution to the minimization problem. Based on (20), we then let g_i be

$$\check{x}_i^{(t)} = \frac{1}{2}\Big(h_i + \hat{x}_i^{(t)} - \sum_{u\in N(i)} J_{iu} \hat{x}_u^{(t)}\Big) \quad \forall i \in V. \qquad (21)$$

The new estimate $\check{x}_i^{(t)}$ is obtained by combining information from neighboring nodes and the node itself. The update expression (21) is just one instance of g_i. In principle, there are many ways to construct the function g_i by building new equalities in terms of x^*.

Upon obtaining the expression for $\check{x}_i^{(t)}$, we study the second minimization (16). Again by plugging (17) into (16), we obtain

$$\begin{aligned}&\tilde{m}_{i\to j}^{(t)}(x_j) \\ &= \kappa + \min_{x_i} \Bigg(\frac{1}{2}\Big[1 - \sum_{u\in N(i)\backslash j} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t+1)}\Big] x_i^2 - \Big[s\check{x}_i^{(t)} \\ &\quad + (1-s)h_i - (1-s)J_{ij}x_j - \sum_{u\in N(i)\backslash j} \tilde{z}_{ui}^{(t)}\Big] x_i \Bigg),\end{aligned}$$

where $\check{x}_i^{(t)}$ is given by (21). We then simplify $\tilde{m}_{i\to j}^{(t)}(x_j)$ by solving the minimization. The resulting expression takes the from:

$$\tilde{m}_{i\to j}^{(t+1)}(x_j) = -\frac{1}{2}(1-s)^2 J_{ij}^2 \tilde{\gamma}_{ij}^{(t+1)} x_j^2 + \tilde{z}_{ij}^{(t+1)} x_j + \kappa',$$

where

$$\tilde{\gamma}_{ij}^{(t+1)} = \frac{1}{1 - \sum_{u\in N(i)\backslash j} (1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}}, \qquad (22)$$

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

and

$$\tilde{z}_{ij}^{(t+1)} = \frac{(1-s)J_{ij}\Big((1-s)h_i + s\check{x}_i^{(t)} - \sum_{u\in N(i)\backslash j} \tilde{z}_{ui}^{(t)}\Big)}{1-\sum_{u\in N(i)\backslash j}(1-s)^2 J_{ui}^2 \tilde{\gamma}_{ui}^{(t)}}, \quad (23)$$

and κ' is a new constant. $\tilde{m}_{i\to j}^{(t+1)}(x_j)$ again takes a quadratic form, which is consistent with (17).

One observes that $\tilde{\gamma}_{ij}^{(t+1)}$ and $\gamma_{ij}^{(t+1)}$ essentially take the same message-passing form. The only difference between them is that $\tilde{\gamma}_{ij}^{(t+1)}$ is derived from $\tilde{J}_s$ while $\gamma_{ij}^{(t+1)}$ is derived from J. Thus as long as s is chosen such that $\tilde{J}_s$ is diagonal dominant or walk-summable, $\tilde{\gamma}_{ij}^{(t+1)}$ always converges.

The parameter $\tilde{z}_{ij}^{(t+1)}$ has an additional term $s\check{x}_i^{(t)}$ compared to $z_{ij}^{(t+1)}$. This additional term $s\hat{x}_i^{(t)}$ is an estimate of sx_i^* in the self-potential (12). If $z_{ij}^{(t)}$ converges, the min-sum-min algorithm then converges to the right solution.

	Stage	Operation
1	*Initialize*	Choose a value s; Set $\tilde{\gamma}_{ij} = 0$ and $\tilde{z}_{ij} = 0$, $\forall (i \to j) \in \vec{E}$
2	*Iterate*	For all $(i \to j) \in \vec{E}$ Update $\hat{x}_i$ using (19) Update $\check{x}_i$ using (21) Update $\tilde{\gamma}_{ij}$ using (22) Update $\tilde{z}_{ij}$ using (23) End
3	*Check*	If $\{\check{x}_i\}, \{\tilde{\gamma_{ij}}\}$ and $\{\tilde{z}_{ij}\}$ become stable, go to 4; else, return to 2.
4	*Output*	Return $\check{x}_i$, $\forall i$.

Table I
MIN-SUM-MIN MESSAGE-PASSING FOR COMPUTING $x^* = \arg\min_x \frac{1}{2}x^TJx - h^Tx$.

Based on the above analysis, we briefly summarize the min-sum-min algorithm for the quadratic minimization (1) in Table I. In the algorithm, we choose the parameter s such that $\tilde{J}_s$ is diagonal dominant. This guarantees that $\tilde{\gamma}_{ij}$ converge for all $(i \to j) \in \vec{E}$.

V. EXPERIMENTAL RESULTS

In the experiment, we test the convergence speed of the min-sum-min algorithm. We study two scenarios: the one where the min-sum algorithm converges and the other one where the min-sum algorithm fails.

We considered two graphs for constructing J as shown in Fig. 1. Graph (a) is a 4-cycle with a chord. Graph (b) is a 5-cycle. For each graph, the matrix J is constructed with its diagonal elements being 1 and its off-diagonal elements being the edge weights as described in the graph. The h vector in (1) for the two graphs are $h = [\begin{array}{cccc} 1 & 2 & 1 & 2 \end{array}]^T$ and $h = [\begin{array}{ccccc} 1 & 2 & 1 & 2 & 1 \end{array}]^T$, respectively.

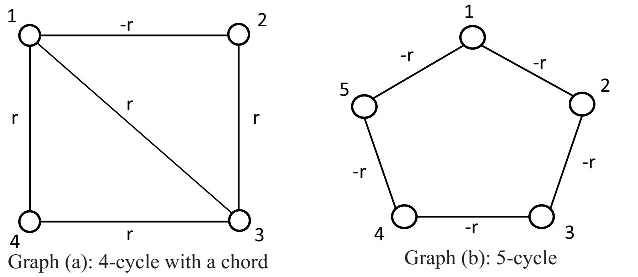


Figure 1. The two graphs for constructing J. The edge weights are as denoted by $-r$ or r in the two graphs.

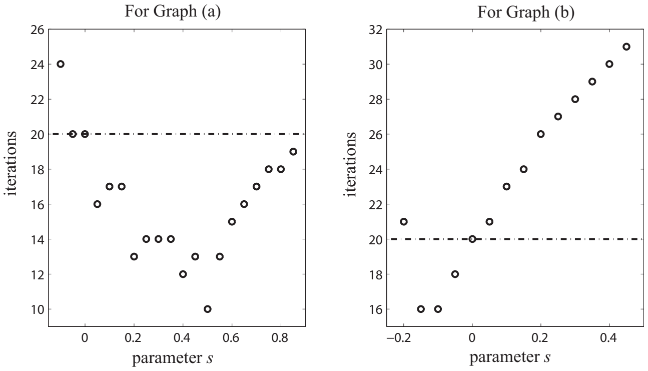


Figure 2. Effect of diagonal loading on the convergence speed. We use the symbol "○" to denote the number of iterations required for different values of s in the min-sum-min algorithm. For comparison, we use the dash-dot line to denote the number of iterations required for the min-sum algorithm.

A. Comparison between the min-sum-min and the min-sum algorithms

In the first experiment, we investigate the scenario where the min-sum algorithm converges. We take the min-sum algorithm as a reference for performance comparison.

We set $r = 0.34$ and $r = 0.4$ in Graph (a) and (b), respectively. Correspondingly, we obtain two realizations for J. The spectral radius of $\bar{J} - I$, where $\bar{J} = [|J_{ij}|]_{i,j=1}^n$, are 0.8709 (for (a)) and 0.8 (for (b)). Thus the J matrices satisfy the walk-summable condition.

In the implementation, we chose the criterion for terminating the algorithm to be $\frac{1}{n}\sum_{i=1}^n |\check{x}_i^{(t)} - x_i^*| \leq 10^{-5}$. We selected the parameter s between -0.2 and 1 for our algorithm.

The experiment results are as shown in Fig. 2. Surprisingly, we observe that for a range of s values, the min-sum-min algorithm outperforms the min-sum algorithm in both cases. The results suggest that there exist more efficient algorithms than the min-sum algorithm. The min-sum-min algorithm is one example in improving the convergence speed. We also tested other values r in Fig 1. The results are similar to those in Fig. 2.

B. Comparison between the min-sum-min and the double-loop algorithms

In the second experiment, we investigate the scenario where the min-sum algorithm fails. We take the double-loop algorithm [11] as a reference for performance comparison.

In this situation, we set $r = 0.45$ and $r = -0.52$ in Graph (a) and (b), respectively. Correspondingly, the spectral radius of $\bar{J} - I$, are 1.1527 (for (a)) and 1.04 (for (b)). The J matrices are not walk-summable anymore.

Again we chose the criterion for terminating the algorithm to be $\frac{1}{n}\sum_{i=1}^{n}|\tilde{x}_i^{(t)} - x_i^*| \leq 10^{-5}$. In implementing the double-loop algorithm, we had to setup one more criterion for the inner-loop iteration. We terminated the inner-loop each time when $\frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i^{(t)} - \hat{x}_i^{(t-1)}| \leq 10^{-5}$.

Fig. 3 and Fig. 4 display the experiment results for Graph (a) and (b), respectively. It is seen from the figures that if our algorithm converges, it converges much faster than the double-loop algorithm. The performance gain in terms of the number of iterations range from hundreds to thousands in the experiment.

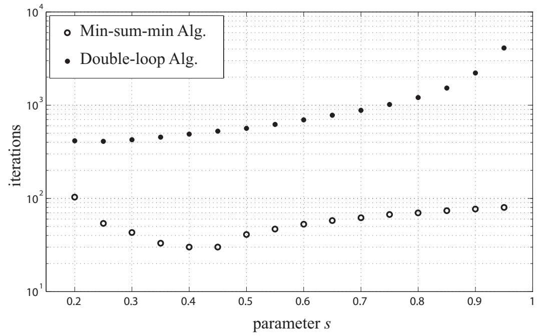


Figure 3. Comparison between the min-sum-min algorithm and the double-loop algorithm for Graph (a).

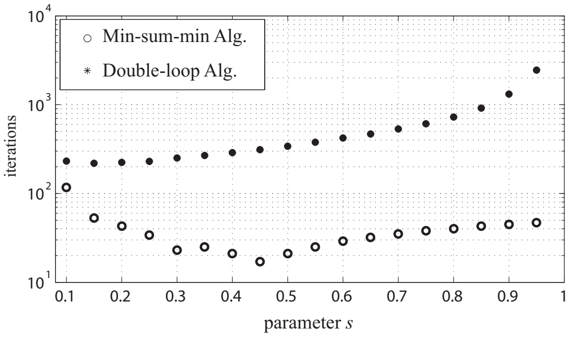


Figure 4. Comparison between the min-sum-min algorithm and the double-loop algorithm for Graph (b).

VI. Conclusion

We have proposed a new min-sum-min message-passing algorithm which includes the min-sum algorithm as a special case. The new algorithm has been derived based on a closed-loop optimization problem which has the same optimal solution as the original problem. Compared to the min-sum algorithm, the min-sum-min algorithm has a free parameter s to choose. This property renders two advantages of the min-sum-min algorithm over the min-sum algorithm. First, the min-sum-min algorithm provides faster convergence speed when the parameter s is chosen properly. Second, in some situations where the min-sum algorithm fails, the min-sum-min algorithm still converges.

One open issue is how to choose the parameter s to make our algorithm most efficient. This issue is quite relevant to engineering in practice.

References

[1] D. Bickson, O. Shental, and D. Dolev, "Distributed Kalman Filter via Gaussian Belief Propagation," in *the 46th Allerton Conf. on Communications, Control and Computing*, 2008.

[2] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," in *Proceedings of the IEEE*, vol. 95, 2007, pp. 1295–1322.

[3] C. C. Moallemi and B. V. Roy, "Consensus Propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, 2006.

[4] A. Montanari, B. Prabhakar, and D. Tse, "Belief Propagation Based Multi-User Detection," in *Proc. 43rd Allerton Conf. on Communications, Control and Computing*, 2005.

[5] P. Rusmevichientong and B. B. Roy, "An analysis of Belief Propagation on the Turbo Decoding Graph with Gaussian Densities," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 745–765, 2001.

[6] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-Sums and Belief Propagation in Gaussian Graphical Models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.

[7] C. C. Moallemi and B. V. Roy, "Convergence of Min-Sum Message Passing for Quadratic Optimization," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2413–2423, 2009.

[8] Y. Weiss and W. T. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," *Neural Computation*, vol. 13, pp. 2173–2200, 2001.

[9] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Walk-sum Interpretation and Analysis of Gaussian Belief Propagation," in *Advances in Neural Information Processing Systems*, vol. 18, Cambridge, MA: MIT Press, 2006.

[10] N. Ruozzi and S. Tatikonda, "Unconstrained Minimization of Quadratic Functions via Min-Sum," in *Proceedings of thee Conference on Information Sciences and systems (CISS)*, March 2010.

[11] J. K. Johnson, D. Bickson, and D. Dolev, "Fixing Convergence of Gaussian Belief Propagation," in *the International Symposium on Information Theory*, 2009.

Parallel Computing the Longest Common Subsequence (LCS) on GPUs: Efficiency and Language Suitability

Amine Dhraief
HANA Research Group
University of Manouba, Tunisia
Email: amine.dhraief@hanalab.org

Raik Issaoui
HANA Research Group
University of Manouba, Tunisia
Email: raik.aissaoui@gmail.com

Abdelfettah Belghith
HANA Research Group
University of Manouba, Tunisia
Email: abdelfattah.belghith@ensi.rnu.tn

***Abstract*—Sequence alignment is one of the most used tools in bioinformatic to find the resemblance among many sequences like ADN, ARN, amino acids. The longest common subsequence (LCS) of biological sequences is an essential and effective technique in sequence alignment. For solving the LCS problem, we resort to dynamic programming approach. Due to the growth of databases sizes of biological sequences, parallel algorithms are the best solution to solve these large size problems. Meantime, the GPU has become an important element for applications that can benefit from parallel computing. In this paper, we first study and compare some languages for parallel development on GPU (CUDA and OpenCL). Then, we present a parallelization approach for solving the LCS problem on GPU. Finally, we evaluate our proposed algorithm on an platform using CUDA, OpenCL and on CPU using the C Language and the OpenMP API. The experiment results show that the implementation of our algorithms in CUDA outperforms the implementation in OpenCL, and the execution time is about 17 times faster on GPUs than on typical CPUs.**

***Index Terms*—Bioinformatic, GPU, Parallel algorithm, LCS, CUDA, OpenCL.**

I. Introduction

Most common studies in the bioinformatic field have evolved towards a more large scale, passing from the analysis of a single gene/protein to the study of a genome/proteome, from a single mechanism within the body towards the biology of the entire system. Hence, it become more and more difficult to achieve these analyses on a single computer, and even for some of them on clusters. The bioinformatic requires now more infrastructure allowing the storage, the transfer of huge amount of data, and finally, the massive computation for their analysis.

Until recently, the CPU, or the computer main chip, dealt most heavy load operations such as physics simulation, bioinformatic, rendering off-line for movies, calculations of risks for financial institutions, weather forecasting, file encoding video and audio [13]. Some of these heavy computations [10] however, are easily parallelizable and can therefore benefit from an architecture for parallel computing. Most parallel architectures were heavy, expensive and targeted at a specific market.

That is until the Graphic Processing Unit (GPU) imposes itself as a major player in the parallel computing [11]. The graphics processors have rapidly evolved, with continual growth performance, more generic architectures and high-level programming tools (CUDA, OpenCL, etc.). GPUs are processors that can run a smaller job a great many times in parallel, while the CPUs are expected to perform lengthy and complex calculations in series [4]. The General Purpose Computation on Graphic Processing Unit (GPGPU) [9] is a new low-cost technology which take advantage of the GPUs to perform massively parallel calculation, traditionally executed on multi-cores CPU.

The GPU computing is designed to accelerate massively parallel algorithms taking advantage of the many execution units present in a GPU. Streaming algorithms are massively parallel algorithms, widely investigated on GPUs. They are algorithms of treatment of data stream in which the input is presented as a sequence of object and can be processed in some passes (sometimes only one). The streaming algorithms are characterized by strongly parallel computations with a little of re-use of input data. These algorithms must be designed to be decomposed in a multitude of small threads that will run in parallel, in groups, on the GPU.

We are talking about thousands of threads, in contrast to the few threads in a CPU. From the software perspective, GPUs are used with more and more tools for developing scientific computing codes using high level programming languages. In front of first spectacular results in terms of reducing the computation time, the major manufacturers now offer professional solutions, dedicated to scientific calculations. Since GPU performance grows faster than CPU performance, the use of GPUs for bioinformatics is therefore a perfect match.

Sequence alignment is a fundamental technique for biologists to investigate the similarity between different species. In computational method, biological sequences are represented as strings and finding the longest common subsequence (LCS) is a widely used method for sequence alignment. Dynamic programming is a classical approach for solving LCS problem, in which a score matrix is filled through a scoring mechanism. The best score is the length of LCS and the subsequence

ISBN: 978-1-61208-161-8

can be found by tracing back the table. The LCS algorithms can be considered as streaming algorithms and thus can be solved using the GPU computing paradigm. In this paper, we investigate to what extend we can solve the LCS problem on GPUs using different hight level programming tools. We show that solving the LCS problem on GPUs has a speed up 17 time faster than solving the LCS problem using traditional parallel API.

The rest of the paper is organized as follows. Section 2 gives an overview and compares the two most used tools in GPU computing, CUDA and OpenCL. Section 3 defines the Longest Common Subsequence problem and presents the used parallelization approach for solving this problem on GPUs. Section 4 shows the experimental results for solving the LCS problem on GPUs and discusses what is the most appropriate developmental environment to solve this problem on a particular GPU device. Section 6 concludes the paper.

II. GPU Computing

Current personal computers can be viewed as multiprocessor stations with shared memory. Thus, developers can easily write parallel programs on these workstation by using specific programming API. OpenMP [6] is one of the most promising parallel programming APIs. It uses a multithreading parallelization method where a single task is divided between several threads that are executed concurrently.

Moreover, most of the current personal computers holds graphic cards which embed several graphical processors. Traditionally, these processors are exclusively used for graphics manipulation purposes. Nonetheless, with the forthcoming of new programming environments, these processors can execute highly parallel programs and intensive calculus. They are consequently called Graphic Processing Units (GPUs) and becoming serious rival to the traditional CPUs.

In the following, we present the two most used used GPUs programming environments, CUDA and OpenCL.

A. CUDA, NVIDIA

The Compute Unified Device Architecture (CUDA) environment developed by Nvidia is a high-performance vector computing environment [3].

On a typical CUDA program, data are first sent from the main memory to the GPU memory. Then, the CPU sends commands to the GPU which performs the computation kernels by scheduling the work on the available hardware. Finally Compute Work Distribution (in GPU) copies the results from the GPU memory to the CPU one via the Host Interface [9].

The hardware architecture used by CUDA consists of a host processor, a host memory and an Nvidia graphics card that supports CUDA. CUDA enabled GPUs are based on the Tesla architecture [9]. These GPUs can run in parallel several thousands of instances (threads) of a unique code (SPMD model). Threads are grouped into blocks which size is defined by the programmer. All threads within a block are executed on a Streaming Multiprocessor (SM) and communicate with each other using a shared memory. Threads in different blocks can not communicate which make the scheduling of the different blocks rapid (independent of the number of SMs used for program execution). In CUDA, a grid is a group of (thread) blocks, with no synchronization between them [8].

The programming language is based on the C language with extensions to indicate whether a portion of the code is executed on the CPU or the GPU. In CUDA, a kernel is a code (usually a function) that can be executed in the the GPU. The process of creating an executable CUDA includes 3 stages associated with the CUDA compiler nvcc. First, the program is divided into CPU section and GPU one's following pragmas inserted by the programmer. The CPU part is compiled by the compiler of the host, while the GPU part is compiled using a specific compiler. The resulting GPU code is a binary CUDA (file Cubin). Both CPU and GPU programs are then linked with libraries that contain CUDA functions for loading the binary code Cubin and send to the GPU for execution.

CUDA has a hierarchy of several types of memory: global, shared, local, texture/constant memory. The global memories are visible to all threads in all blocks, the biggest and the slowest. The shared memories are visible to all threads in a particular block, a medium size and an average speed of communication. The local memories are only visible to a particular thread, the smallest and the fastest. The texture/-constant memories are visible to all threads in all blocks and they are read-only memory. Each thread can read/write independently in registers and local memory. All threads in the same block can read/write (communicate) in shared memory. And all threads in the same grid can read/write (communicate) in global and constant memory.

B. OpenCl, Apple

OpenCL (Open Compute Language) is an open API dedicated to massively parallel computing, initially developed by Apple and the Khronos Group -a consortium of firms engaged in the development of open APIs like OpenGL.

The CUDA is a non-free proprietary software and CUDA programs can exclusively be executed on Nvidia GPUs. Apple has proposed OpenCL to exploit the power of GPUs without being locked in a range of products from a particular manufacturer (Nvidia). OpenCL can therefore be seen as an API intended to standardize the GPU computing. OpenCL is a common language to all architectures, it is not intended only to GPUs but also CPUs and covers accelerators such as the Cell (in the Playstation 3).

OpenCL basic functions are exactly the same as for CUDA: a kernel is sent to the accelerator (compute device) which is composed of ”compute units” whose ”processing elements” working on ”work items”.

Finally, CUDA compatible cards (GeForce, Quadro, Tesla) support both CUDA and OpenCL.

C. Comparison

Table I presents a qualitative comparison between CUDA and OpenCL. This comparison highlights the fact that CUDA is a more mature technology than OpenCL; whereas, OpenCL has the merit to be an open standard. While CUDA can be used only on Nvidia‘s GPU, OpenCL can be used on a wide range of GPU devices.

	CUDA	**OpenCL**
Technologies	Owner	Open
Start	2006	2008
Free SDK	Yes	Depend on vendor
Many vendors	No,only NVIDIA	Yes: Apple,AMD, IBM
Multiple OS	Yes;Windows, Linux, Mac OS X;32 and 64 bits	Depend on vendor
Heterogeneous devices	No, Only NVIDIA GPUs	Yes

TABLE I: QUALITATIVE COMPARISON BETWEEN CUDA AND OPENCL

From a memory management point of view, in both CUDA and OpenCL, the host and the device memories are separated. The device memory are hierarchical designed and must be explicitly controlled by the programmer. The memory model of OpenCL is more abstracted and supplies more way for various implementations. CUDA explicitly defines all the memories levels, whereas in OpenCL, these details are hidden as they are device dependent.

CUDA and OpenCL are similar in several aspects. They are concentrated on data parallel computing model. They provide a C-basic language customized for device programming. The device, the execution scheme and the memory models are very similar.

Most of the differences result from differences of their origin. CUDA is the technology owner of Nvidia and targets only Nvidia devices. OpenCL is open and targets several devices. CUDA has been on the market earlier than OpenCL (2006 vs. 2008) and has more support, applications, related research and products. Finally, CUDA is more documented than OpenCL.

III. The LCS problem

The extraction of the longest common subsequence of two sequences is a current problem in the domain of datamining. The extraction of theses subsequence is often used as a technique of comparison to get the similarity degree between two sequences.

A. Definition

A sequence is a finished suite of symbols taken in a finished set. If $U = \langle a_1, a_2, .., a_n\rangle$ is a sequence, where $a_1, a_2, .., a_n$ are letters, the integer n is the length of u. A sequence $V = \langle b_1, b_2, .., b_n\rangle$ is a subsequence of $U = \langle a_1, a_2, .., a_n\rangle$ if there are integers $i_1, i_2, .., i_m (1 \leq i_1 < i_2 < .. < i_m \leq n)$ where $b_k = a_{i_k}$ for $k \in [1, m]$.

For example, $V = \langle B, C, D, B\rangle$ is a subsequence of $U = \langle A, B, C, B, D, A, B\rangle$. A sequence W is a subsequence common to sequences U and V if W is a subsequence of U and of V. A common subsequence is maximal or is a longer subsequence if it is of length maximal. For example: sequences $\langle B, C, B, A\rangle$ and $\langle B, D, A, B\rangle$ are the longest common subsequences of $\langle A, B, C, B, D, A, B\rangle$ and of $\langle B, D, C, A, B, A\rangle$.

B. Parallel algorithms for the LCS problem

The dynamic programming is a classical approach for solving the LCS problem. It is based on the filling of a score matrix through a scoring mechanism. The best score is the length of the LCS and the subsequence can be found by tracing back the table.

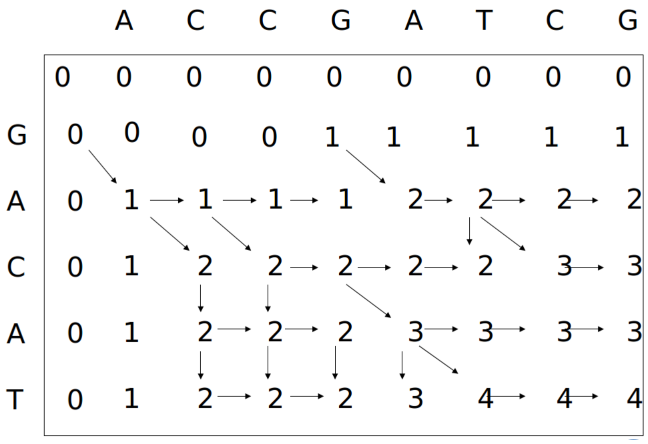
Let m and n be the lengths of two strings to be compared. We determine the length of a maximal common subsequence in $A = \langle a_1, a_2, .., a_n\rangle$ and $B = \langle b_1, b_2, .., b_m\rangle$.

We note $L(i, j)$ the length of a maximal common subsequence in $\langle a_1, a_2, ..a_i\rangle$ and $\langle b_1, b_2, .., b_j\rangle (0 \leq j \leq m, 0 \leq i \leq n)$.

$$L(i,j) \begin{cases} 0 \quad if\, i = 0\, or\, j = 0 \\ L(i-1, j-1) + 1 \quad if\, a_i = b_j \\ max(L(i, j-1), L(i-1, j)) \quad else. \end{cases} \tag{1}$$

We use the above scoring function to fill the matrix row by row (Fig. 1).

The highest calculated score in the score matrix is the length of the LCS. In Fig. 1, the length is 4. In the scoring matrix, the LCS is traced back from the highest score point (4) to the score 1.



		A	C	C	G	A	T	C	G
	0	0	0	0	0	0	0	0	0
G	0	0	0	0	1	1	1	1	1
A	0	1	1	1	1	2	2	2	2
C	0	1	2	2	2	2	2	3	3
A	0	1	2	2	2	3	3	3	3
T	0	1	2	2	2	3	4	4	4

Fig. 1: Example of filling the LCS matrix score.

The time and space complexity of this dynamic programming approach is $O(mn)$, where m and n are the length of the two compared strings. Several parallel algorithms in the literature target to solve the LCS problem. In the following table, we present the parallel complexity and the number of processors of some algorithms (m and n are the length of the two compared strings, p is the number of available processor):

Reference	Parallel Complexity	Number of processors
Apotolico et al. 1990 [1]	$O(logm * logn)$	$mn/logm$
Lu and Lin 1994 [7]	$O(logm + logn)$	$mn/logm$
Babu and Saxena 1997 [2]	$O(logm)$	mn
Xu et al. 2005 [12]	$O(mn/p)$	p
Krusche et al. Tiskin 2006[5]	$O(n)$	p

C. The parallelization approach

In the scoring matrix that dynamic program algorithm constructs according to the equation Eq. 1, for all i and j $(1 \leq i \leq n, 1 \leq j \leq m)$, $L[i,j]$ depends on three entries; $L[i-1,j-1], L[i-1,j]$ and $L[i,j-1]$ (as shown in Fig. 2). In other words, $L[i,j]$ depends on the data in the same row and the same column. So the cells in the same column or same row cannot be computed in parallel.

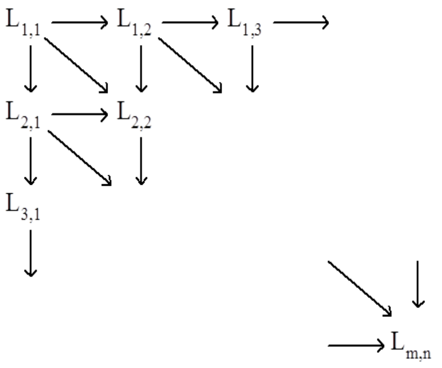


Fig. 2: Data dependency in the score matrix of a dynamic programming algorithm for solving LCS problem.

We start computing $L_{1,1}$ then $L_{1,2}$ and $L_{2,1}$ in the same time, afterthat $L_{1,3}$, $L_{2,2}$ and $L_{3,1}$. We continue until we fill the matrix. We notice that it is possible to compute cells in the same anti-diagonal parallelly. To parallelize the dynamic programming algorithm, we have to compute the score matrix in the anti-diagonal direction (see Fig. 3).

IV. Results

A. Performance measurement methodology

We implement our LCS algorithm using CUDA, OpenCL, OpenMP and the C language on a computer equipped with a processor Intel(R) Core(TM) Quad CPU Q9400 2.66GHZ, a random access memory 8 GBytes and a graphics board NVIDIA GT 430. We use as operating system Ubuntu 10.04. The characteristics of our graphics board are listed in the following:

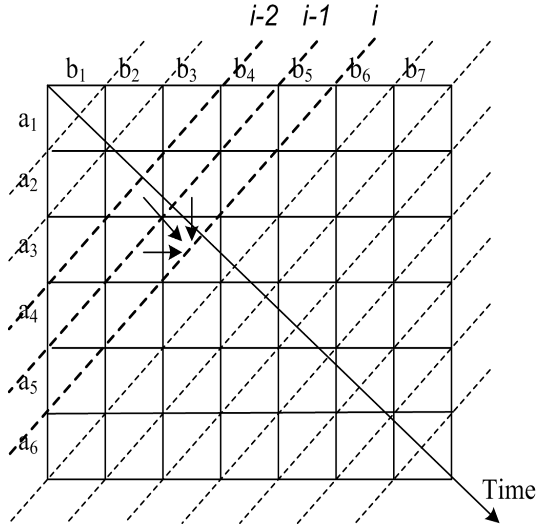


Fig. 3: The parallelization approach

- GPU Engine Specifications:

CUDA Cores	96
Graphics Clock(MHz	700
Processor Clock(MHz)	1400

- Memory Specifications:

Memory C lock(MHz)	800-900 (DDR3)
Standard Memory Config	1GB DDR3
Memory Interface Width	128-bit
Memory Bandwidth(GB/sec)	25.6-28.8

We measure the filling of the LCS scoring matrix runtime without taking into account the initialization of sequences A and B. The unit of measure of the runtime is the millisecond. In all the following tests, we run as many trials as needed to reach a 95% confidence interval at $\varepsilon = 1\%$ of the average value.

B. The execution time

Fig. 4 illustrates the execution time of the four implementations of our LCS algorithm (CUDA, OpenCL, OpenMP and C) versus the size of the two sequences.

We can clearly see that for the C implementation of our algorithm, whenever we increase the size of the compared sequence the execution time exponentially increases. This behavior is justified by the absence of parallelization in the C implementation of our algorithm.

Both CUDA, OpenCL and OpenMP implementations of our algorithm use the parallelization approach presented in section III-C. Thus, the execution time of these three versions of our algorithm increases linearly as we increase the size of the compared sequence

ISBN: 978-1-61208-161-8

For long sequences, we notice that the fastest version of our algorithm is the CUDA one, and specifically more than the OpenCL implementation. As CUDA is the SDK developed by NVIDIA for their graphic processors, it is then more appropriate for NVIDIA devices than OpenCL.

For short sequences, we plot in Fig. 5 the execution time of the four implementations of our algorithm using a logarithmic scale in the y-axis.

Fig. 5 shows that, for short sequences, OpenMP outperforms OpenCL and has similar performances than CUDA. In fact, CPU is more performant than GPU for small problems where parallelism has a negative effect on the execution time. The thread management slacken the overall execution time and signficanlty exceeds the kernel execution time.

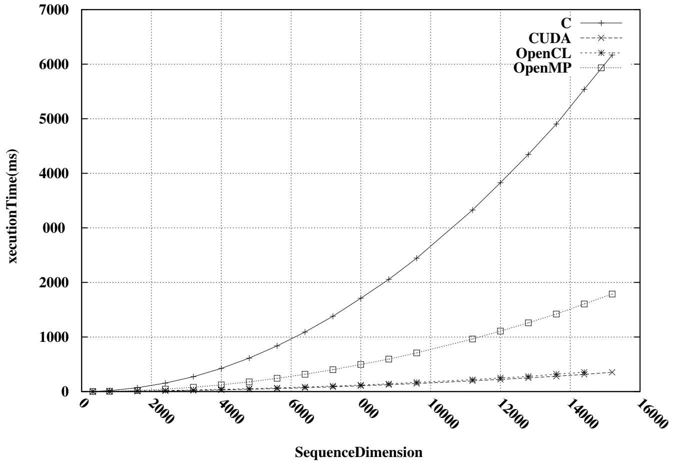


Fig. 4: The LCS execution time

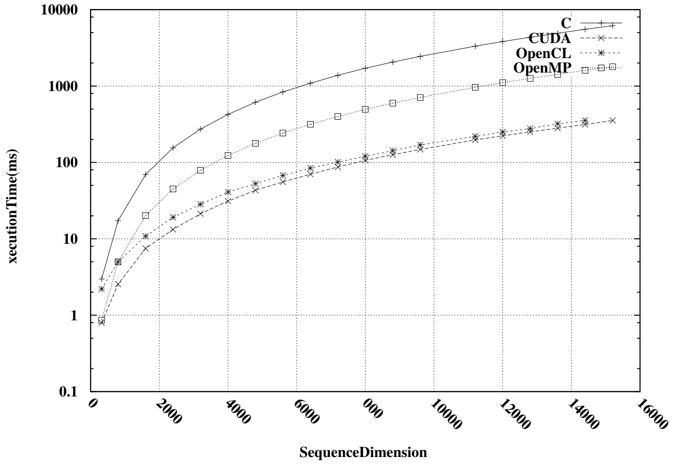


Fig. 5: The LCS execution time (logarithmic scale)

Fig. 6 compares the latency of the kernel runtime and the latency taken to transfer the data to the main memory (RAM). Only OpenCL allows us to determine these latencies.

Foremost, we notice that the required time to perform the memory transfer is the most important in the operation of filling of the score matrix. In addition, the memory transfer latency increases with the size of the sequences.

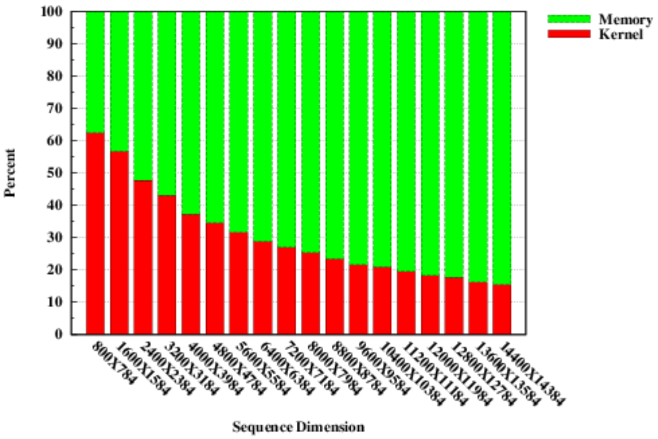


Fig. 6: Kernel latency vs. memory transfer latency

C. The speed up

In the parallel computing, the speed up shows to what extent a parallel algorithm is faster than a corresponding sequential algorithm.

Analytically, we define the speed up as:

$$\text{SpeedUp} = \frac{\text{Sequential execution time}}{\text{Parallel execution time}} \quad (2)$$

Fig. 7 shows the speed up of the implementation versions of our algorithms in OpenMP, OpenCL et CUDA versus the size of the two sequences.

The measured speed up for the implementation of our algorithm in Open Mp (on CPU) is bounded at 3.22 times. In fact, we use OpenMP to get advantage of the parallelism provided by the Quad core CPU. But, because of the other operating system threads, the speed up cannot achieve 4 times

The implementation of our algorithms in CUDA outperforms the implementation in OpenCL, as we are using an NVIDIA graphic card and CUDA is specifically designed for the NVIDIA devices.

Our results show that for long sequences, the implementation of our algorithm in CUDA (on GPU) is 17 times faster than the one on CPU.

This result is explained by the CUDA architecture. In CUDA, threads are grouped in blocks (the programmer chooses their size). All threads in the same block are executed on the same Streaming Multiprocessor (SM) and communicate via a shared memory.

Our parallelism approach is based on the filling of a score matrix in the anti-diagonal sense. The threads are filling the matrix by calculating anti-diagonals one by one, every anti-diagonal uses the necessary number of block to calculate it in parallel.

We notice that up to a certain sequence size (800 characters), the GPU is under exploited. This is mainly due to the management of threads which delays the execution time. The CPU can handle with this problem effeciently for small sequences.

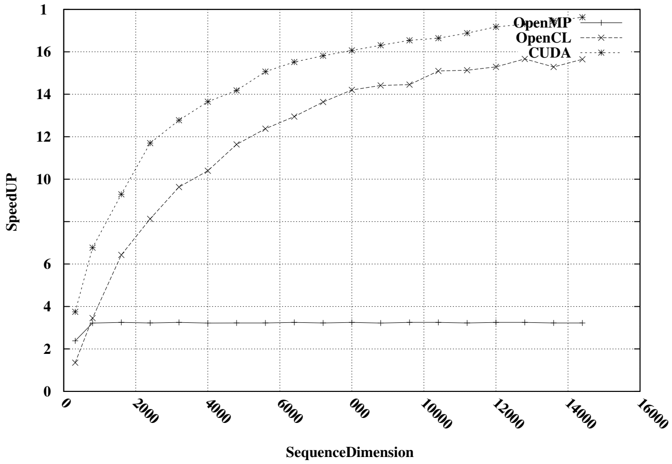


Fig. 7: Speed UP

V. Conclusion

In bioinformatic, biological sequences are represented as strings and finding the longest common subsequence (LCS) is a widely used method for sequence alignment.

We have focused in this paper on the parallelization of a dynamic programming algorithm for solving the LCS problem. For this purpose, we have studied some languages for parallel development on GPU (CUDA and OpenCL). Then, we have presented a parallelization approach for solving the LCS problem on GPU. We have evaluated our proposed algorithm on an NVIDIA platform using CUDA, OpenCL and on CPU using the C Language and the OpenMP API.

The results have shown that the studied algorithm enables higher degree of parallelism and achieves a good speedup on GPU. In addition, during this experiment, we have proved that CUDA is more suitable for NVIDIA devices than OpenCL. Finally, we have demonstrated that the major contribution in the execution time is the latency of the memory transfer from GPU Global memory to CPU RAM.

References

[1] Alberto Apostolico, Mikhail J. Atallah, Lawrence L. Larmore, and Scott McFaddin. Efficient parallel algorithms for string editing and related problems. *SIAM J. Comput.*, 19:968–988, September 1990.

[2] K. Nandan Babu and Sanjeev Saxena. Parallel algorithms for the longest common subsequence problem. In *Proceedings of the Fourth International Conference on High-Performance Computing*, HIPC '97, pages 120–, Washington, DC, USA, 1997. IEEE Computer Society.

[3] Michael Garland, Scott Le Grand, John Nickolls, Joshua Anderson, Jim Hardwick, Scott Morton, Everett Phillips, Yao Zhang, and Vasily Volkov. Parallel Computing Experiences with CUDA. *IEEE Micro*, 28:13–27, July 2008.

[4] Qihang Huang, Zhiyi Huang, Paul Werstein, and Martin Purvis. GPU as a General Purpose Computing Resource. In *Parallel and Distributed Computing, Applications and Technologies, 2008. PDCAT 2008. Ninth International Conference on*, pages 151–158, 2008.

[5] Peter Krusche and Alexandre Tiskin. Efficient longest common subsequence computation using bulk-synchronous parallelism. In *ICCSA (5)*, volume 3984 of *Lecture Notes in Computer Science*, pages 165–174. Springer, 2006.

[6] Seyong Lee, Seung J. Min, and Rudolf Eigenmann. OpenMP to GPGPU: a compiler framework for automatic translation and optimization. *SIGPLAN Not.*, 44:101–110, February 2009.

[7] Mi Lu and Hua Lin. Parallel algorithms for the longest common subsequence problem. *IEEE Trans. Parallel Distrib. Syst.*, 5:835–848, August 1994.

[8] David P. Luebke. Cuda: Scalable parallel programming for high-performance scientific computing. In *ISBI*, pages 836–838, 2008.

[9] John D. Owens, Mike Houston, David Luebke, Simon Green, John E. Stone, and James C. Phillips. GPU Computing. *Proceedings of the IEEE*, 96(5):879–899, May 2008.

[10] James C. Phillips and John E. Stone. Probing biomolecular machines with graphics processors. *Commun. ACM*, 52:34–41, October 2009.

[11] John Shalf. The new landscape of parallel computer architecture. *Journal of Physics Conference Series*, 78(1):012066, 2007.

[12] Xiaohua Xu, Ling Chen, Yi Pan, and Ping He. Fast parallel algorithms for the longest common subsequence problem using an optical bus. In *Computational Science and Its Applications ICCSA 2005*, volume 3482 of *Lecture Notes in Computer Science*, pages 91–115. Springer Berlin / Heidelberg, 2005.

[13] Zhiguo Xu and Rajive Bagrodia. GPU-Accelerated Evaluation Platform for High Fidelity Network Modeling. In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation*, PADS '07, pages 131–140, Washington, DC, USA, 2007. IEEE Computer Society.

MPI-based Solution for Efficient Data Access in Java HPC

Aidan Fries* † ‡, Jordi Portell* † ‡, Yago Isasi* † ‡, Javier Castañeda* † ‡, Raül Sirvent§, and Guillermo L. Taboada¶
*Department of Astronomy and Meteorology, University of Barcelona, Barcelona, (Spain)
†Institute for Space Studies of Catalonia (IEEC), Barcelona, (Spain)
‡Institute of Cosmos Sciences (ICC), Barcelona, (Spain)
afries@am.ub.es, jportell@am.ub.es, yisasi@am.ub.es, jcastapo@am.ub.es
§BSC-CNS Barcelona Supercomputing Center, Barcelona, (Spain). raul.sirvent@bsc.es
¶Computer Architecture Group, University of A Coruña, A Coruña (Spain). taboada@udc.es

***Abstract*—Efficient data access is extremely important for many applications in HPC. In many cases, processes running in one node will need to access data held in another node, as well as access data held in some central storage device. In I/O-intensive applications, accessing data not held in the local node can become a bottleneck, especially in cases where the remotely stored data is accessed repeatedly, and when accessing data from virtual machines such as in Java. To address this issue, we have designed and implemented a data cache system, which offers efficient data access to Java applications in HPC. This system, which we call MPJ-Cache, makes use of a Java-based message-passing implementation, such as F-MPJ, and it provides a high-level API for the accessing of data. MPJ-Cache can improve the performance of I/O operations for certain Java applications in HPC by reducing significantly the I/O overhead. In this paper, we describe MPJ-Cache, including the data communication layer, as well as the caching features of the system, and we show how it can be used to improve I/O performance for HPC applications. The comparative performance evaluation of this system against the file system of the MareNostrum supercomputer (Barcelona Supercomputing Center) has shown important performance benefits. Finally, we also show the impact of this solution on a challenging problem such as the data processing system for the ESA Gaia space mission.**

***Keywords*-Java Communications; Data Cache; F-MPJ; Gaia; GPFS; Myrinet**

I. Introduction

A typical distributed-memory HPC environment includes many computing nodes, each node containing one or more processors and each processor containing one or more cores. Each node may be connected to every other node over some high-speed, low latency network, while each node may also be connected to a central storage device.

In recent years, the most significant trends in distributed-memory HPC environments have included the move towards a larger number of cores per processor, an increased awareness of the issue of power consumption and a massive growth in the volume of data being handled. The increase in the number of available cores will typically lead to an increase in the number of parallel processes running in a computing node, and therefore an increase in the number of processes accessing data. These factors combine to make the issue of efficient data access extremely important.

In the case of I/O-intensive applications, where the processes running in the computing nodes may need to access data stored in the shared storage device, I/O can become a significant factor affecting the overall performance of the application. The importance of I/O efficiency increases with the number of parallel processes, and the problem is further amplified if the data is accessed repeatedly.

MPI continues as the leading approach for implementing inter-process communication in distributed memory environments, offering point-to-point as well as collective communication functions. However, despite the strengths and maturity of MPI, writing applications that can take full advantage of the resources of a distributed environment, such as a cluster of computing nodes, can be quite difficult. In the case of complex applications, where there may be strong dependencies between processes running in different nodes, the programming effort required to implement the communication can be considerable and is often bug-prone.

The Client-Server model is a long established and intuitive approach for making data available to a group of consumers. In order to avoid the potential bottleneck issue associated with many processes accessing a shared storage device, we designed a data cache system, which we call MPJ-Cache. This system involves the execution of Server processes in some of the available nodes, which maintain a cache of data; while the communication between the Clients and the Servers is built on top of an implementation of Message Passing in Java (MPJ), and can take advantage of any high-speed network support provided by the underlying MPJ application.

Gaia [1] is a European Space Agency mission, whose primary objective is to chart a 3D map of around one billion stars in our Galaxy. The Data Analysis and Processing Consortium (DPAC) is the organisation with responsibility to process the Gaia data. It is a policy within DPAC that all software should be written in Java. The selection of Java for this kind of large, scientific, data processing project is relatively uncommon. Therefore, the Gaia data processing represents an opportunity to study the use of Java to implement a scientific processing pipeline, and its execution in a HPC environment. In this paper, we will discuss two DPAC applications, both of which are I/O-intensive and could benefit from the use of MPJ-Cache.

The first of these applications is GAia System Simulator (GASS), which simulates the raw telemetry stream that will be generated by the satellite during its mission lifetime. Secondly, we will discuss Intermediate Data Updating (IDU), which is one of the applications that will process real Gaia data.

The rest of this paper is organised as follows. In Section II we give a general summary of the current status of Java in HPC, and in particular, on the status of Java-based data communication in HPC. In Section III, we describe the I/O problems faced by I/O-intensive Java applications in HPC, specifically the issue of I/O bottlenecks. Our solution to the aforementioned problem is presented in Section IV, where we briefly describe the communication layer of MPJ-Cache, as well as the caching features of the system. In Section V, we describe a set of tests designed to compare the performance of our data cache against direct access of GPFS. The results of these tests are given in Section VI. Finally, in Section VII, we give our conclusions, and mention some further work that we intend to carry out in this area.

II. Related work

Despite the continuing popularity of Java in general computing, and the many independent projects which have developed extensions and libraries for Java applications in HPC, the use of Java in HPC and by the scientific communities remains relatively low. We conducted an investigation into the developments for Java in HPC over the last 15 years. We looked at papers published, as well as libraries and tools that were developed to aid the use of Java in HPC. It is clearly evident that there was a very significant level of work in this area, roughly speaking, during the period 1999 to 2003. This period corresponds with the activity of the Java Grande Forum [5], a initiative amongst the Java scientific community to investigate possible additions to the Java language, and to encourage its use in HPC and scientific communities. However, since then, the number of papers and projects in this area has reduced significantly, and the number of projects which are actively developing or supporting libraries for Java in HPC now appears quite low.

It is almost certainly the case that part of the reason for the slow adoption of Java in HPC is simply due to the inertia of moving from the languages which have traditionally being used in the HPC and scientific computing communities such as Fortran and C/C++. It is also due to the initial reputation that Java acquired as providing poor performance due to it being an interpreted language. Finally, part of the problem may also be a lack of reliable and supported HPC-specific Java libraries in areas such as data communication.

COMP Superscalar (COMPSs) [4] is a runtime environment which allows for the automatic parallelisation of serial applications, for their execution in a HPC environment. This process involves COMPSs analysing the application; identifying tasks of a certain granularity within that application; determining the dependencies between these tasks; generating a task graph; and where possible, executing tasks in parallel. COMPSs also manages the input and output for each task. It is a powerful tool, allowing seemingly serial applications to become parallel, and taking advantage of the available HPC resources. However, it does not deal with the potential bottleneck associated with many processes accessing a shared storage device. Although COMPSs removes the issue from the concern of user applications, COMPSs itself may encounter I/O issues if it tries to distribute some data to a large number of nodes. Also, there are circumstances when application developers may prefer to maintain control of the actual flow of data around the available hardware. Therefore, we believe that another solution, dealing with the potential I/O bottleneck and providing application developers with a high-level, HPC-specific data-access API would be a useful contribution to Java in HPC.

Based on our investigation into the available libraries which provide data communication to Java applications in distributed memory environments, the 3 most commonly used options are: Sockets, RMI, and Message Passing. In this work we decided to focus our work on the MPJ approach as it has been reported to provide the highest performance in HPC applications on low latency networks [7]. MPJ can be implemented in a number of ways, including the use of Java RMI, Java Native Interface (JNI) — to call an underlying native message passing library, and also through the use of Java sockets. Each approach has advantages and disadvantages in the areas of efficiency, portability and complexity. The use of RMI assures portability, as it is a pure Java solution. However, it may not be the most efficient solution in the presence of any high speed communication hardware, which RMI might not take full advantage of. The use of JNI allows for the efficient use of high-speed networks using native libraries, however it has portability problems. Finally, the implementation of MPI using Java sockets requires a considerable development effort. Fortunately, we have identified implementations of MPJ: MPJ Express [6] and F-MPJ [8], which are being actively developed and supported. MPJ Express and F-MPJ both implement the same specification of MPJ — `mpiJava` 1.2 [2] — so it is easy to swap between these implementations.

III. The problem

An important trend in HPC is the move towards an ever increasing number of cores per processor, and consequentially an increase in the number of processes that are typically executed per computing node. The volume of data that these processes must handle is also increasing. Despite the availability of high performance storage devices and networks, the accessing of data held in shared storage devices can act as a bottleneck in the processing of data, if there are a large number of processes accessing the data and if it is accessed repeatedly.

The objective of this work is to find an efficient and scalable mechanism that allows for a large number of processes running in separate computing-nodes to be able to access some remote data, where the I/O performance achieved is minimally affected by the number of processes accessing the data.

A. *Gaia data processing in MareNostrum*

The following two applications, both of which are part of the Gaia data processing task and will run on the MareNostrum supercomputer at the Barcelona Super Computing Center (BSC), represent different use-cases where the I/O problem described in the previous section emerges as an issue.

1) GASS: During the simulation of the telemetry stream, GASS has to process over 1 billion stellar sources. These simulations require the use of many worker processes (thus far, simulations involving over 3000 cores working simultaneously have been executed). These worker processes need to repeatedly access a set of shared data-files during the execution of GASS (such as spectra and instrument calibration files). If all of the executing worker processes simply access these files directly on the GPFS, then its performance decreases to unacceptable I/O response times.

2) IDU: IDU itself is composed of several independent tasks, which run in a particular sequence and which are effectively independent serial applications. However, some of these tasks have dependencies on the output from other tasks, and the input for one task may come from the output of another task which was executed in a different computing node. Therefore data transfers are required between the execution of each task in order to deliver the correct input data to the correct process in the correct computing node.

IDU forms part of an iterative chain of processes that will process the Gaia data. This chain of processes will be executed once every 6 months over a 5 year period. Each time that IDU is executed it will process all of the data amassed at that point, so as the mission continues, the volume of data will increase reaching roughly 100TB at the end of the mission. Although the actual volume of data may not be overwhelming, the challenging aspect of the processing is the relationships within the data, and the reorganising and movement of the data, which must be carried out between tasks.

B. *Scalability tests of GPFS*

An initial version of IDU involved all processes reading their input data from the GPFS storage device. In order to determine how well this approach would scale to a large number of worker processes running in a large number of computing-nodes, we carried out some scalability tests. The input files for these tests were of equal size and represented a certain amount of processing. In all cases, we just ran one worker process per node.

We found that if we increased the number of IDU worker processes, and therefore the number of processes accessing the GPFS, the system scaled fairly well up to about 8 nodes — the speed-up factor being 7.4, whereas the speed-up factor for 16 nodes was just 13, as illustrated in Fig. 1. Thus, although GPFS reveals a rather good scalability, the I/O performance per node starts decreasing significantly already at just 16 nodes.

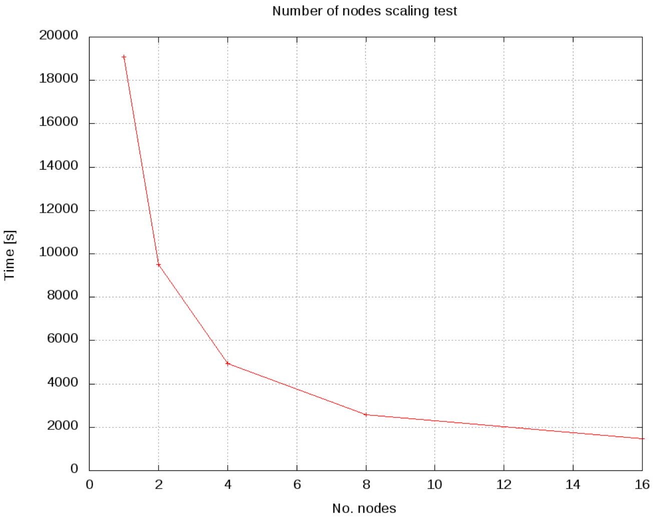


Fig. 1. Processing time of an IDU prototype over a fixed amount of data when using 1 to 16 nodes accessing directly the GPFS disk

IV. THE SOLUTION — MPJ-CACHE

We have designed a scheme which involves the grouping of the available computing nodes into Node Groups (NGs). Within each NG, one node is designated as the Node Group Manager (NGM). A Server process is executed in the NGM, which creates and maintains a cache of data that user application processes can query. We refer to user application processes simply as Client processes. The inter-node communication — between Clients and Servers, as well as amongst Clients — is implemented on top of an implementation of MPJ, making use of any high-speed network support provided by the MPJ implementation.

There are, of course, a number of possible configurations that a given set of nodes can be grouped into. For example, 64 nodes can be grouped into 4 groups of 16, or into 8 groups of 8 nodes. One of the objectives of the tests described in the next section was to determine the optimal NG configuration in order to maximise data access performance for a given set of files and a given application.

MPJ-Cache has two relatively distinct components within it, namely the the communication layer and the data cache. These components are illustrated in Fig 2.

A. *MPJ-Cache - the communication layer*

The objective of the MPJ-Cache communication layer is to provide user applications with a high-level API of data access methods useful in HPC environments, while making best use of the available communication resources such as any high-speed, low-latency networks which might be present. MPJ-Cache makes use of an implementation of MPJ to perform inter-process communication. Implementations of MPJ, such as MPJ Express and F-MPJ implement an API of methods, such as `MPI_Sendrecv()`. MPJ-Cache builds upon the MPJ API, and offers its own API

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-161-8

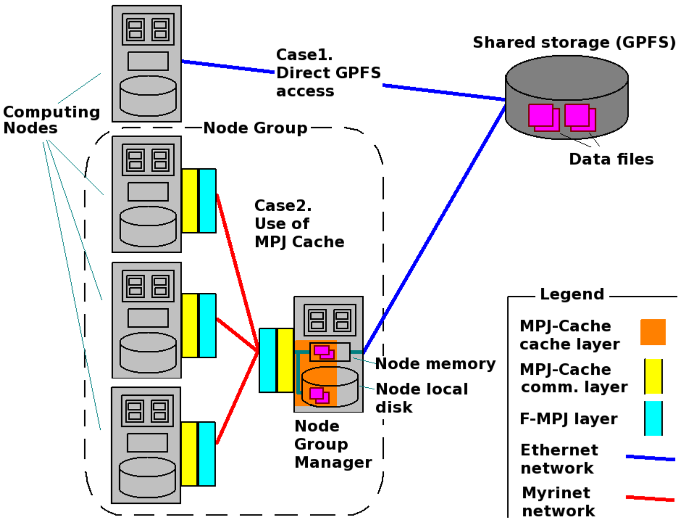


Fig. 2. MPJ-Cache components and test setup

of methods which are at a higher level of abstraction. For example, the MPJ-Cache API includes the static method `retrieveSingleFileAsByteArray()`, which allows Clients to request a file as an array of bytes. The implementation of this method involves 2 calls to the MPJ method `MPI_Sendrecv()`. Firstly, a request is made to find the size of the file. Then, once the size of the file is known, a buffer can be allocated that will contain the data and a second call to `MPI_Sendrecv()` is made requesting the actual file. However, from the perspective of the Client application, it must only make one call to `retrieveSingleFileAsByteArray()`.

The communication layer also includes file splitting and recombining functionality, which allows accessing files with sizes that exceed the maximum data size permitted by the underlying implementation of MPJ. Such large files are split into smaller chunks by the Server, which are then sent in separate messages, and finally, once all of the chunks have been received at the Client side, they are recombined and passed to the Client application.

B. MPJ-Cache - the cache

In the context of MPJ-Cache, the cache refers to the part of the Server application that maintains the actual cache of data. The Server can be configured to either store the most frequently used data in memory, on the local disk of the node that it is running on, or to simply act as a gateway, retrieving data from a remote location as requests are received. Indeed, the cache can be spread over these 3 locations. The Server maintains a list of all of the files that it is aware of: those it has in memory, those that it has on its local disk and those which might be in a remote location. The Server initializes its cache at start-up, based on its configuration, but it can update the cache during the execution of the application, depending on its configuration. A number of policies to control the maintenance of the cache including First In, First Out (FIFO) and Least Recently Used (LRU) are available.

C. Selection of MPJ Implementations

Among the available MPJ implementations, MPJ Express and F-MPJ are the libraries with a more active development, so they have been evaluated in order to select the best performer for use by MPJ-Cache. F-MPJ implements support for several interconnection networks, among them the support for InfiniBand in the low level communication device `ibvdev`, which runs directly on top of IBV (InfiniBand Verbs), and support on Myrinet in the device `omxdev`, which runs directly on top of MX. We first carried out initial tests to determine their performance in our particular production environment. MPJ Express, F-MPJ, MPICH-MX and MX utilities and were tested using a Pingpong and a Broadcast benchmark. The results of these tests are given in Tables I and II. These showed that F-MPJ performs very well in the target environment, in particular, it offers very low latency for short messages and good bandwidth with longer messages.

TABLE I
PINGPONG TESTS (POINT-TO-POINT COMMUNICATIONS)

Application	Latency (μms)	Bandwidth (MB/s)
F-MPJ	17.0	246.12
MX utilities	17.0	247.0
MPJ Express	23.2	180.8
MPICH-MX	17.0	247

TABLE II
BROADCAST TEST - 8 NODES, 1 PROCESS PER NODE

Data	MPICH-MX		F-MPJ		MPJ Express	
	Latency (μms)	BW (MB/s)	Latency (μms)	BW (MB/s)	Latency (μms)	BW (MB/s)
2MB	21.9	96.0	25.8	81.3	43.6	48.1
4MB	42.7	98.2	52.4	80.0	87.5	47.9

D. MPJ-Cache in practice

One consideration that users of MPJ-Cache must take into account is that applications wishing to make use of the system must be executed within the MPJ environment. This has an effect on how the user application can be launched. The approach that we took is to initially launch the same class — `LaunchAppsInProcesses` — in all of the MPJ processes. Within the MPJ environment each process is identified by a unique identifier called the process rank. We followed the approach that the process with rank 0 would act as a Server process, while instances of the user application would be launched in the other processes. In order to create several NGs, with each NG containing a Server process and a number of Client processes, we simply need to launch several jobs, each one starting an instance of the MPJ environment.

V. THE TESTS

In order to directly compare the performance of MPJ-Cache against direct access to GPFS, we executed a campaign of tests, designed to reflect the characteristics of real applications running in a HPC environment. These tests can be grouped into 2 main categories. In the first case, the Client processes retrieve data directly from the GPFS, while in the second case the Clients retrieve the same data using MPJ-Cache. These 2 cases are illustrated in Fig 2. Tests involving the use of MPJ-Cache can be further categorised into those involving 1 NG and those involving multiple NGs.

In order to be representative of real applications, the data communication within the tests was interleaved with "processing" by the Client applications — simulated by "sleeps" of the Client processes between each request for data. Tests involving sleeps of varying lengths were carried out. Each Client also performs an initial sleep during its initialisation to ensure that not all of the Clients begin requesting data at the same time.

In total, 96 tests were executed in this test campaign. In all cases, the tests involved Clients retrieving 20 different files. There are many test parameters which were varied including the size of the data files (1MB, 10MB, 100MB), the number of clients (16, 32, 64 or 128), the number of NGs (1, 4 or 8), and the sleep time. In the case of the MPJ-Cache tests, the Servers were configured to store all of the data in a cache in memory.

The BandWidth (BW) referred to in these results has been calculated using the time a request takes to be processed from the Clients perspective. Therefore, it includes any delay which might occur at the Server side.

A. System features

The MareNostrum Supercomputer [3] consists of 2560 JS21 blade computing nodes, each with 2 dual-core IBM 64-bit PowerPC 970MP processors running at 2.3 GHz, which totals 10240 cores (9.2 GFlops per core), and 8 GB memory per node. The MareNostrum was ranked 5th of the world according to the Top500 List at the time of installation (2006), although currently it is ranked 118th based on its Linpack performance (measured 64 TFlops out of 94 TFlops peak). MareNostrum nodes are interconnected through low latency Myrinet 2000 network (12 switches conform the fabric), as well as through Gigabit Ethernet, this latter used to access the 280 TB of disk storage though GPFS (General Parallel File System). The OS is SuSE Linux Enterprise Server 9, the JVM is IBM J9 1.6.0, the F-MPJ release is 0.1.0 and the MPI implementation is MPICH-MX 1.2.7.4, while the MX driver version is 1.2.7-64.

VI. Results

A. Direct GPFS access vs. MPJ-Cache with 1 NG

MPJ-Cache generally out performs GPFS in those tests with smaller files sizes and a short sleep between requests, as shown in Table III, and illustrated in Fig 3.

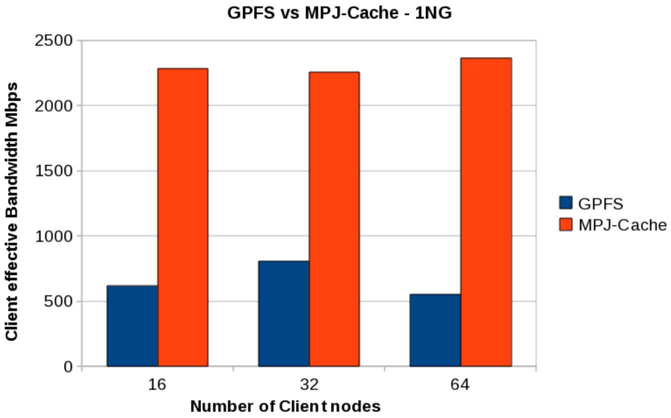


Fig. 3. Comparison of GPFS vs MPJ-Cache when using 1 NG on small files (1MB) and frequent requests (1ms)

TABLE III
MPJ-Cache Results — small files, small sleep, 1NG

Clients	Data	Sleep (ms)	GPFS BW (Mbps)	Cache BW (Mbps)	Speed-up
16	1MB	1	616	2284	3.7
32	1MB	1	806	2256	2.8
64	1MB	1	553	2365	4.3

MPJ-Cache also outperformed GPFS in tests with large file sizes and a long sleep period, as shown in Table IV, the one exception being the case of 128 nodes, where GPFS performed best. The explanation for the poor performance of MPJ-Cache in that test is that the Server process was overloaded. In other words, the Server was receiving requests faster than it was able to handle them, therefore, a queue of requests built up.

In fact, in most of the cases where the GPFS outperformed MPJ-Cache, the difference was explained by an overloading of the Server. We confirmed this by examining the Client log files in those cases where the MPJ-Cache performed poorly. We noted that the initial requests received by the Server were processed quickly, and therefore the initial bandwidth experienced by the Clients was relatively good. However, as more Clients began to request data from the Server, the Server became overloaded and the average reply time experienced by the Clients increased — hence the decrease in the calculated bandwidth.

B. Direct GPFS access vs. MPJ-Cache with multiple NGs

With this set of tests we intend to find the optimal configuration for accessing data from a given set of nodes. Effectively we want to maximise the total amount of data that can be transferred around the system during a given period of time. We call this rate the "aggregate data rate". We must note that this data rate contains within it, the initial sleep as well as

TABLE IV
MPJ-Cache Results — large files, large sleep, 1NG

Clients	Data	Sleep (ms)	GPFS BW (Mbps)	Cache BW (Mbps)	Speed-up
16	100MB	20000	1026	1639	1.6
32	100MB	20000	1090	2164	2.0
64	100MB	20000	265	295	1.1
128	100MB	20000	796	83	0.1

the inter-request sleeps performed by the Clients in order to simulate real applications.

The highest aggregate data rate achieved by direct access of GPFS was 13.7Gbps, achieved through the use of 128 Client nodes, requesting 100MB files with a sleep of 200 milliseconds between requests. Interestingly, if the sleep was reduced to 10 milliseconds, the data rate fell to 9.1Gbps.

The highest aggregate data rate achieved through the use of MPJ-Cache was 103Gbps. This was also achieved using 128 Client nodes, but arranged in 8 NGs. Therefore, 136 nodes in total were used (including 8 MPJ-Cache Servers). The files used were 100MB each, while the sleep was 1 millisecond.

The results of this test campaign showed, that given a certain number of nodes, MPJ-Cache allows for a much higher aggregate data rate than direct GPFS access, as illustrated in Fig. 4

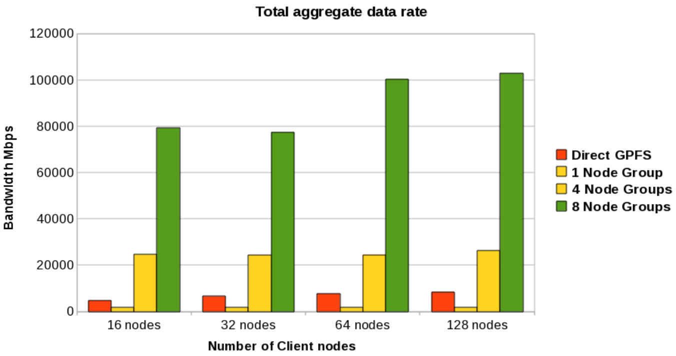


Fig. 4. Total aggregate data rate for GPFS and MPJ-Cache, using different Node Group configurations. 100MB files have been used in this case.

VII. Conclusion and Future Work

Our tests confirmed the impressive performance for high-speed, low-latency networks achievable with F-MPJ. MPJ-Cache, configured to create a single NG, offers better performance than GPFS for small numbers of Client nodes, and especially when working with small files and frequent requests. We observed in our test campaign that the available bandwidth on the Myrinet network was optimally being used, and the performance of MPJ-Cache only decreases when the Server process is overloaded with requests. Furthermore, when we move to the situation of multiple NGs, the total aggregate data rate obtainable through the use of MPJ-Cache is much higher than that obtainable for the same number of nodes directly accessing GPFS.

Although our data cache system is a relatively thin layer, sitting on top of an implementation of MPJ, we believe that there are a range of applications which could benefit from its use, not just the applications described in this paper. The creation of data caches amongst the available computing nodes can avoid the I/O bottleneck which can occur when many processes are accessing a central storage device, while the use of F-MPJ allows for the best performance to be extracted from the available network.

In the case of HPC environments which are shared by many users, one possible issue is how the applications of one user may affect the applications of other users. One of the benefits of using MPJ-Cache is that, due to the caching, there will be a reduced number of accesses of the shared disk, which should improve the disk I/O performance experienced by other users.

We intend to improve the functionality of MPJ-Cache by adding more data access methods to the API that it offers, including the addition of methods to support a “push” model from Server to Client, in addition to the existing “pull” methods which allow for the Client-Server model. Finally, we intend to test MPJ-Cache in other HPC environments, such as its use with the LUSTRE and Panasas file systems, as well as to try to identify other Java-based I/O-intensive applications, running in HPC environments, which could benefit from the use of MPJ-Cache.

Acknowledgment

This work was supported by the MICINN (Spanish Ministry of Science and Innovation) - FEDER through grant AYA2009-14648-C02-01 and CONSOLIDER CSD2007-00050. We also acknowledge the support that the F-MPJ project receives from MICINN, TIN2010-16735.

References

[1] European Space Agency. The gaia mission. http://gaia.esa.int/ [Last visited: July 2011].

[2] B. Carpenter, G. Fox, S.-H. Ko, and S. Lim. mpijava 1.2: Api specification. http://www.hpjava.org/reports/mpiJava-spec/mpiJava-spec/mpiJava-spec.html [Last visited: July 2011].

[3] Barcelona Supercomputing Center. Marenostrum supercomputer. http://www.bsc.es/plantillaA.php?cat_id=200 [Last visited: July 2011].

[4] M. Danelutto, P. Fragopoulou, V. Getov, E. Tejedor, R. M. Badia, T. Kielmann, and V. Getov. A component-based integrated toolkit. In *Making Grids Work*, pages 139–151. Springer US, 2008.

[5] M. Philippsen, R. F. Boisvert, V Getov, R Pozo, J. E. Moreira, D. Gannon, and G. Fox. Javagrande - high performance computing with java. In *Proceedings of the 5th International Workshop on Applied Parallel Computing, New Paradigms for HPC in Industry and Academia*, 2001.

[6] A. Shafi, B. Carpenter, and M. Baker. Nested parallelism for multi-core hpc systems using java. *Journal of Parallel and Distributed Computing*, 69(6):532–545, 2009.

[7] G. L. Taboada, J. Touriño, and R. Doallo. Java for high performance computing: Assessment of current research and practice. In *Proc. 7th International Conference on the Principles and Practice of Programming in Java (PPPJ’09)*, pages 30–39, Calgary, Alberta, Canada, 2009.

[8] G. L. Taboada, J. Touriño, and R. Doallo. F-mpj: Scalable java message-passing communications on parallel systems. *Journal of Supercomputing (In press, DOI: 10.1007/s11227-009-0270-0)*, 2011.

Real-Time Processing and Archiving Strategies

Isabel Schnoor
ITS Public Sector
IBM Germany
Hamburg, Germany
schnoor@de.ibm.com

***Abstract*— This paper will discuss how real-time processing is affecting archiving strategies. This puts up the thesis, what type of system is needed to ensure write & retrieval of real-time data from an archive infrastructure. The paper will not focus on compliance, rather on archiving strategies in context of system requirements and availabilities for real-time data applications and hence where real-time processing is applicable.**

Keywords- Archive; Real-time; long-term archiving; HSM; HPSS; real-time compression

I. Introduction

Today due to huge data growth, more digitization needs, mobile devices producing more data, faster supercomputer power, and applications, end user expect to store and archive their data in real-time and have continuously real-time access to that data. Keeping up with this fast growing demand of user behavior of "self-service" real-time data access, common and established IT-infrastructures are put under stress. Storing everything on disks becomes to large to backup, leading to long back-up windows, and increasing energy cost for more power consumption of disks systems. Especially, if needed to store data at least for 10 years or longer.

Ideal to store large amounts of data long-term are tape technologies. Due to the availability of open standards with Linear Tape-Open (LTO) as magnetic tape data storage technology, originally developed in the late 1990s, it is accepted to archive long-term data. Tape-based archival systems suffer from poor random access performance, which prevents the use of inter-media redundancy techniques and auditing, and requires the preservation of legacy hardware [1]. Many disk-based systems are ill-suited for long-term storage because their high energy demands and management requirements make them cost-ineffective for archival purposes [1].

However, combinations of disk and tape storage infrastructures do not fulfill the requirement of fast write and retrieval access to data at adequate speed, as today end user require "real-time" processing and availability. De-duplication is reducing the amount of data even further, because de-duplication technology is now common place in the backup market [2]. But backup is not the same as archiving. An archive is a collection of computer files that have been packaged together for backup, to transport to some other location, for saving away from the computer so that more hard disk storage can be made available, or for some other purpose [3]. An archive can include a simple list of files or files organized under a directory or catalog structure (depending on how a particular program supports archiving) [3].

Section two will first define real-time processing and archives as well as the terminus data transfer rate (DTR). It will position, what possible storage solutions could be used for high volume and near real-time or real-time processing applications. As software for hierarchical storage management for very large data volumes, High Performance Storage System (HPSS) is described. Section three will evaluate how HPSS is being used for real-time applications. Section four closes with a short summary and future outlook of HPSS and other storage solution for archiving.

This paper will position archiving strategies to real-time processing needs and discuss how in a context of an extreme scale archive environment, e.g., HPSS, is applicable for real-time applications.

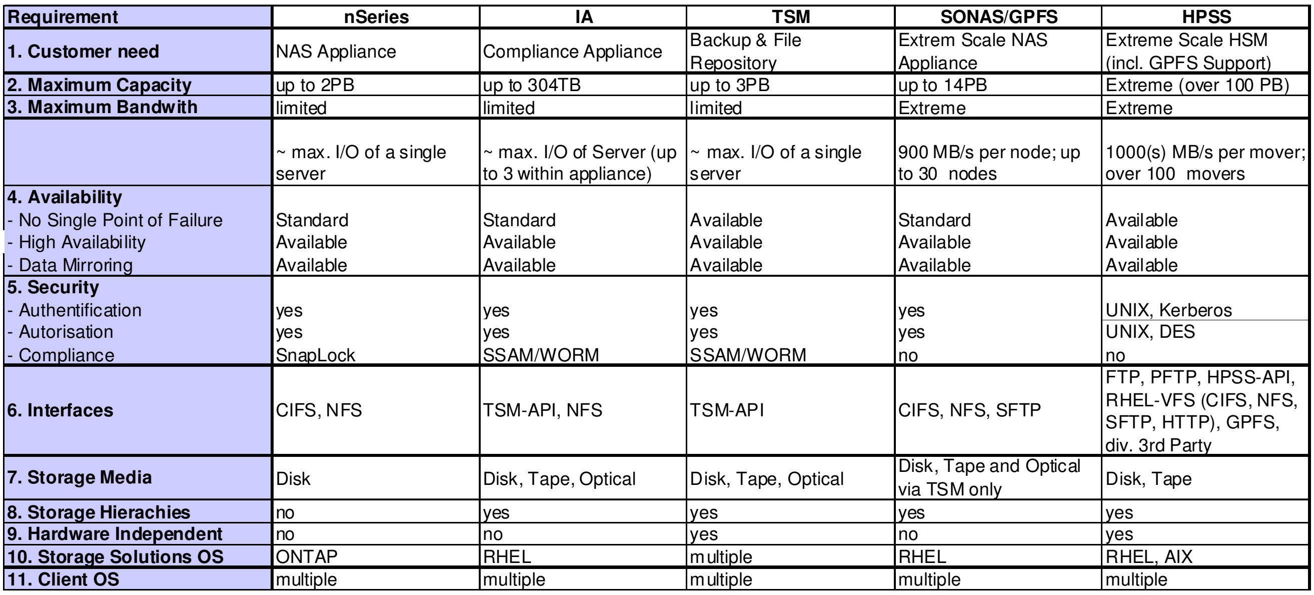
II. Archiving Strategies for real-time applications

Real-time processing is defined according to Wikipedia as follows: "Each transaction in real-time processing is unique. It is not part of a group of transactions, even though those transactions are processed in the same manner. Transactions in real-time processing are stand-alone both in the entry to the system and also in the handling of output" [4].

Archives can be differentiated by types of deployment of storage components, i.e., Disks, Tape, Hierarchical Storage Management Software, or Appliances. All of these components can be positioned in entry, midrange or enterprise storage systems, and being enriched with so-called "Storage Optimizers", i.e., SAN Volume Controller (SVC), Easy Tier, or IBM Real-Time Compression Appliances. Archiving strategies are rather defined by their usage requirements either for compliance need, regular batch processing backup and archive (i.e., for databases, applications systems like SAP, Oracle, etc.) or for long-term archiving, including clustered HSM file repositories. One of the available HSM clustered file repositories is High Performance Storage System (HPSS) for extreme scale environments.

ISBN: 978-1-61208-161-8

Figure 1. Positioning of IBM Archiving Solutions, IBM Corp. / IBM Germany, J.A. Kerr, O.J. Knopf, and I. Schnoor



Requirement	nSeries	IA	TSM	SONAS/GPFS	HPSS
1. Customer need	NAS Appliance	Compliance Appliance	Backup & File Repository	Extrem Scale NAS Appliance	Extreme Scale HSM (incl. GPFS Support)
2. Maximum Capacity	up to 2PB	up to 304TB	up to 3PB	up to 14PB	Extreme (over 100 PB)
3. Maximum Bandwith	limited	limited	limited	Extreme	Extreme
	~ max. I/O of a single server	~ max. I/O of Server (up to 3 within appliance)	~ max. I/O of a single server	900 MB/s per node; up to 30 nodes	1000(s) MB/s per mover; over 100 movers
4. Availability - No Single Point of Failure - High Availability - Data Mirroring	 Standard Available Available	 Standard Available Available	 Available Available Available	 Standard Available Available	 Available Available Available
5. Security - Authentification - Autorisation - Compliance	 yes yes SnapLock	 yes yes SSAM/WORM	 yes yes SSAM/WORM	 yes yes no	 UNIX, Kerberos UNIX, DES no
6. Interfaces	CIFS, NFS	TSM-API, NFS	TSM-API	CIFS, NFS, SFTP	FTP, PFTP, HPSS-API, RHEL-VFS (CIFS, NFS, SFTP, HTTP), GPFS, div. 3rd Party
7. Storage Media	Disk	Disk, Tape, Optical	Disk, Tape, Optical	Disk, Tape and Optical via TSM only	Disk, Tape
8. Storage Hierachies	no	yes	yes	yes	yes
9. Hardware Independent	no	no	yes	no	yes
10. Storage Solutions OS	ONTAP	RHEL	multiple	RHEL	RHEL, AIX
11. Client OS	multiple	multiple	multiple	multiple	multiple

IBM has a broad range of storage systems available, addressing various needs for storage infrastructures, i.e., for open systems and optimized for z/OS and System i, for block-based or file-based storage infrastructures. The strategy of the IBM System Storage products is aligned to the need of delivering the right system by "fit for purpose" for any workload. For archiving needs, the figure 1 shows a table, outlining by eleven requirements, the possible archive solutions from IBM. The following products and solutions are: IBM NSeries, IBM Information Archive (IA), IBM Tivoli Storage Manager and Tivoli Storage Manager for Space Management (TSM/HSM), Scale-out Network Attached Storage with IBM General Parallel File System (SONAS/GPFS), and HPSS. According to type of archiving need and scenario, figure 1 describes what storage solution is applicable and possible to select.

Researchers get the brightest ideas and with better equipment to analyze, research, discover, and exploit, the possibilities of new ways of developing fundamental experiments or create new simulations is large. Due to this type of research, high amounts of volumes of data are generated and need to be stored and archived in large file systems. Real-time applications are common in weather prediction, measuring seismic activity, applications for simulations for the new development of airplanes, in air traffic control systems, or in rail way switching systems etc. Perhaps, in future due to new compression mechanism, i.e., IBM Random Access Compression Engine (RACE) can reduce the Terabytes (TB) managed or even Petabytes (PB) in archives. Hence, making it easier, that even NAS environments, become applicable for long-term archiving. But for certain industries, i.e., for basic research organizations or governmental institutions, the need is clear, according to sources, to grow in data by far more than 60 PB by 2016 [5]. Real-time compression may come in use for unstructured files and in hence in use for NAS environments. Relating this to archives, real-time compression in primary storage may help to reduce the amount of data to be archived, i.e., of databases, IBM Lotus Notes®, Text, CAD, or VMware VMDK files [6], with a compression rate up to 80 percent.

It is of interest that from high performance computing and extreme scale environments, often new and innovative results can be derived for commercial IT environments. As a recent study by the German Federal Ministry of Education and Research shows, 43 percent of businesses are participating in recent research for developments in HPC-Software for scalable parallel computing for new algorithms in Germany [7]. HPSS is at current in use by more than 30 sites worldwide, mainly in use at governmental institutions, research organisations or in defence, but rarely in place at commercial businesses.

In order to measure how fast data from applications is running, is defined by its data transfer rate to the storage infrastructure. In a world where programs and files are becoming ever-larger, the highest data transfer rate is most desirable [8]. However, as technology moves quickly to advance the data transfer rate of many components, consumers are often faced with systems that incorporate varying specifications [8]. This circumstance becomes visible in figure 1.

Figure 2 explores the simple idea, how to position each storage solution by its data volume to be archived into a single system ("high" as in greater than 15 PB, "low" as below 1 PB data volume) and by which favorable processing type ("batch" vs. "real-time"). This positioning does not account the type of network used and assume a standard file-size of ~ 160 MB [9]. This is an average file size for an typical HPSS installation.

Quadrant A positions the storage systems and solutions that apply mostly for real-time processing and are able to store per single system more than 1 PB. Quadrant B positions storage systems and solutions that are rather batch oriented as processing type of data. However, newer features in metadata management of the utilized relational databases, will make these solutions more applicable for real-time processing. Quadrant C positions storage systems and solutions that have a lower possibility to store less than 1PB and are rather seen as "Storage Optimizers" and hence not for long-term archiving needs applicable storage solutions for high data volumes. Quadrant D positions furthermore NAS filer storage systems that have a high real-time processing capability, however lack the vertical scalability in storage capacity. In addition, RACE is an appliance in front of the Storage Area Network (SAN) and not used as a single storage system, rather reflecting a "Storage Optimizer" as earlier described.

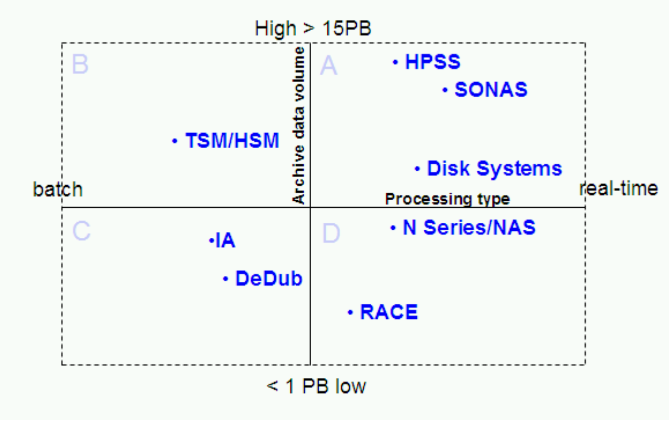


Figure 2. Positioning Archive Systems by Data Volume vs. Processing Type (batch vs. real-time)

Data transfer rate (DTR), is the speed at which data can be transmitted between devices. This is sometimes referred to as throughput [8]. The data transfer rate of a device is often expressed in kilobits or megabits per second, abbreviated as kbps and mbps respectively. It might also be expressed in kilobytes or megabytes, or KB/sec and MB/sec [8].

HPSS is software that manages PB of data on disk and robotic tape libraries. HPSS provides highly flexible and scalable hierarchical storage management that keeps recently used data on disk and less recently used data on tape. HPSS uses cluster, LAN and/or SAN technology to aggregate the capacity and performance of many computers, disks, and tape drives into a single virtual file system of exceptional size and versatility [10]. This approach enables HPSS to easily meet otherwise unachievable demands of total storage capacity, file sizes, data rates, and number of objects stored [10]. HPSS is known to be the leading archival storage software system to fulfill extreme scale requirements [11]. Speeds are limited only by the underlying computers, networks, and storage devices. HPSS can manage parallel data transfers from multiple network-connected disk arrays at hundreds of megabytes per second. These capabilities make possible new data-intensive applications such as high definition digitized video at rates sufficient to support real-time viewing [12]. At the German Climate Computing Center (DKRZ), the implemented HPSS as the data archive has available bidirectional bandwidth of 3 GigaByte/s (sustained), and 5 GigaByte/s (peak) [13]. A central technical goal of HPSS is to move large files between storage devices and parallel or clustered computers at speeds many times faster than today's commercial storage system software products [14].

HPSS supports striping to disk and tape. At 100 MB/s, it takes almost 3 hours to write a 1 TB file to a single tape. Using an 8-way tape stripe, that time is cut to less than 25 minutes [14]! Commercially available 2,5'' disks based on MLC chips, solid state storage featuring SATA (3 Gb/sec.) interface, have data transfer rates of 150 MB/sec. (read) and 90 MB/sec. (write) [15]. It must be differentiated in reading data and writing data to demonstrate real-time or near real-time data storage access. Writing data to storage is sometimes called "data ingestion" [15]. A typical day of writing data to the archive at the European Centre for Medium-range Weather Forecast (ECMWF) is ~ 42 TB [17]. Reading from the archive is ~ 19TB per day [17].

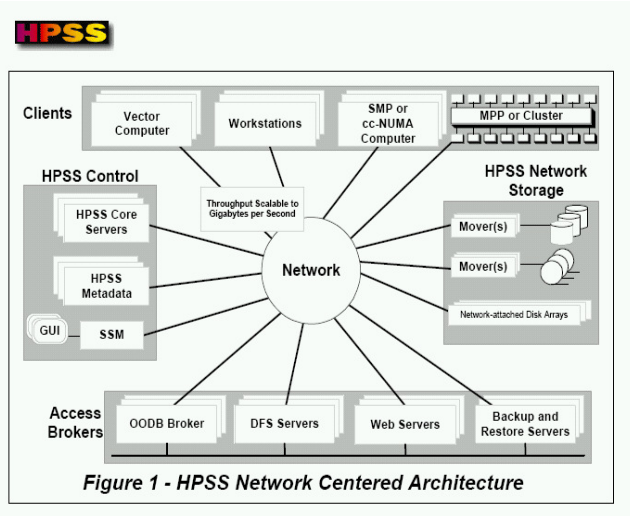


Figure 3. Basics of High Performance Storage System [12]

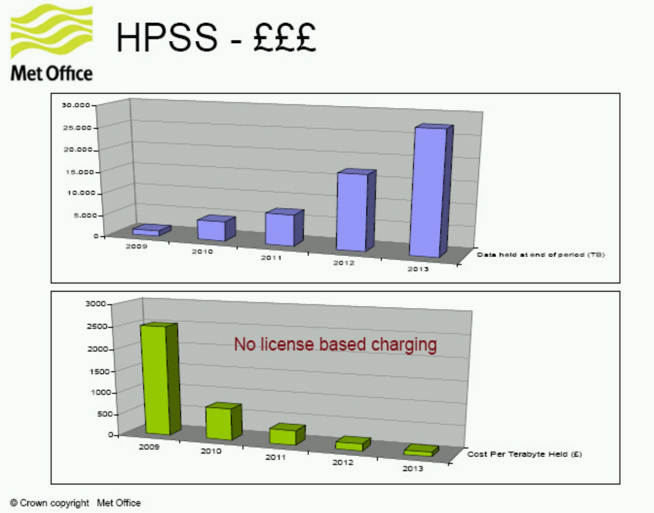
The focus of HPSS is the network, not a single server processor as in conventional storage systems. HPSS provides servers and movers that can be distributed across a high performance network to provide scalability and parallelism. The basis for this architecture is the IEEE Mass Storage System Reference Model, Version 5 [12], see Figure 3. Once a transfer session is established, the actual data transfer takes place directly between the client and the storage device controller [12].

HPSS achieves high data transfer rates by eliminating overhead normally associated with data transfer operations.

In general, HPSS servers establish transfer sessions but are not involved in actual transfer of data. For network-attached storage devices supporting IPI-3 third party transfer protocols, HPSS Movers deliver data at device speeds. For example, with a single HiPPI attached disk array supporting IPI-3 third party protocols, HPSS transfers a single data stream at over 50MB/sec [12].

The HPSS Application Program Interface (API) supports parallel or sequential access to storage devices by clients executing parallel or sequential applications. HPSS also provides a Parallel File Transfer Protocol. HPSS can even manage data transfers in a situation where the number of data sources and destination are different. Parallel data transfer is vital in situations that demand fast access to very large files [12].

Due to the fact that HPSS does not have a volume-based licensing, the invest for such a high performance archiving solution is costly for the first year, but after it operates, the pay-off is clear, as it only costs for the tape library and media have to be calculated in future years. At the UK Met Office the payoff of HPSS for research data was clear: the archive size was in 2010 approx. 3PB, but with a forecast to year 2013 by 20PB [18], the UK Met Office needed a solution that support Extreme Scale Computing, store and keep data as central asset for research; but at the same time to be Energy efficient and affordable long-term. The use of HPSS is hence technically and commercially attractive for this type of storage and archiving demands see Figure 4.


HPSS - £££

© Crown copyright Met Office

Figure 4. HPSS - £££ [18]

III. REAL-TIME PROCESSING WITH HPSS

The recommended use of HPSS is for storing data sets that require high availability and that need longer-term storage. It is not intended for real-time access or short-term storage for temporary data [18]. Therefore, the team “near real-time” for HPSS is introduced by Harry Hulen, IBM Global Services – Federal, Houston, Texas [16]. He describes the possibility of using HPSS for a near real-time application, as follows:

“A good example of near-real-time data ingestion is when a scientific experiment is running and producing data that must be captured, such as a high energy physics experiment or a satellite that is sending down data that must be ingested and put on tape. There are two cases: "bursty" data that requires high rate ingestion for a period of time, and then stops for a while. In this case a disk cache usually takes care of the real time part of the problem, and then data migrates to tape when the burst of ingestion is complete. A tougher case is when there is a steady stream of data for a long period of time. In this case the tape system must operate at close to real time for a long period. An example of this kind of data would be the satellite that continually downlinks data.

Usually the read side of the problem is defined by response time and not data rate. If data is on disk and the storage system is not suffering from queuing, then the response is quick. The best example to think of is Yahoo or Google. If data has migrated to tape and then purged from disk, then the user may have to wait for minutes instead of seconds or sub-second responses. This reading of data is real-time, or near real-time, in the sense that the answer closely follows the interrogation; however, the terms "real time" and "near real time" are used less often for this type of transaction. A distributed system like HPSS are valuable for high data rate near real time data ingestion because a cluster can keep data flowing to more disks and tapes than a monolithic, single-computer storage system can manage.

When designing a tape system for near-real-time storage, it is necessary to accommodate the non-real-time characteristics of tape, particularly what happens when a tape is full. At that point the tape cartridge must be unmounted and another tape mounted. This process can take two to three minutes in a modern robotic tape library. There must be sufficient tape drives so that other tapes are waiting and ready to go, or there must be sufficient disk to catch and hold the ingested data. In any case, it is necessary to over-provision a near real time system when compared with a system that can gracefully allow longer queues to develop.

HPSS supports near-real-time data ingestion for both bursty data and steady state data with its distributed data mover architecture and its ability to write stripes of data across multiple tape drives.”

Within HPSS, file metadata are maintained in DB2, IBM's real-time database system [19]. Further, other major application domains, such as real-time data collection, also require such extreme scale storage. We believe the HPSS architecture and basic implementation built around a scalable relational database management system (IBM’s DB2) make it well suited to this challenge [20].

IV. CONCLUSION

Real-time processing and long-term archiving needs will not be possible to be accommodated at the same time in just one single storage solution. Either for the main objective to support really real-time processing by high I/O systems with their respective DTRs disks systems are applicable or for long-term archiving needs with tape infrastructures. A combination of disks and tape systems, i.e., with HPSS, will address at leas near real-time processing needs.

There is to take into consideration future developments and features, such as a) Real-time compression mechanism and b) future of HPSS:

a) IBM Real-Time Compression: IBM Random Access Compression Engine (RACE) is IBM patented technology that is the key to IBM Real-time Compression Appliances for NAS. RACE technology allows read and write operations from any location within a compressed file while avoiding the need to decompress the whole file. Additionally, the technology preserves all file metadata as it is stored on disk, making the compressed file transparent to the application. The technology enables data compression without compromising performance or data integrity. By operating in real time and reducing the amount written to disk, IBM Real-time Compression solutions enable enhanced storage performance and efficiency [21].

Ideal for seismic research, the company Halliburton has developed a storage solution called “PetroStor^{TM}” that is an integrated solution comprised of technologies from Landmark, NetApp and Storwize that can lower the cost of on-line archival storage to less than $1,000 per TB by using industry leading technology for data compression and de-duplication for oil and gas exploration organizations [22].

b) HPSS will include in future real-time applications requirements. Already in 2004, at SC’04, IBM demonstrated with HPSS performance using three computers, one each for HPSS, reading and writing. A large 128 GB file was written and read in 512 MB blocks using16-way striped SAN-attached disk files, using 8 host bus adapters on each client computer. As one computer wrote each block, it was immediately read by a second computer, thus demonstrating "read behind write" performance. The file transfers were measured at 1016 MB/s on the write side and 1008 MB/s on the read side, for an aggregate data rate of just over two GB per second [23].

For instantaneous throughput, the HPSS development aims at ~ 50GB/s for HPSS [23]. In addition, there are multiple developments underway, to how to most effectively utilize DB2 partitioning and other capabilities to support multiple dynamic Core Servers (Metadata Servers). The upcoming newest Version 8 of HPSS will include architecture of Distributed Metadata. Using the DB2’s Data Partitioning Feature this will provide the necessary infrastructure to distribute HPSS metadata. The partitioning feature is based on a share nothing architecture, where each system manages the local partition, but has access to all partitions transparently [24]. DB2 is extremely well tested and supported by IBM; HPSS development and takes advantage of this mature and robust capability [24].

First prototypes at the HPSS development team have shown the new architecture provides 10x performance of HPSS V7 single metadata server architecture [24]. Therefore, HPSS will accommodate both, end user expectation to have data in real-time or at least “near real-term” access, store them long-term, and, for IT-Managers, HPSS demonstrates a viable alternative disk and tape solution scenario, to save costs for extreme scale future storage data growth needs.

As discussed in this paper, real-time processing needs will be vital to be addressed soon in data management infrastructure and technology decisions.

V. Acknowledgment

The author would like to thank Harry Hulen from IBM Houston, Texas, for his help in getting the examples for read and write and the real-time implication for HPSS.

References

[1] TechRepublic Pro, (2008) Source: NetApp, Pergamum: Replacing Tape With Energy Efficient, Reliable, Disk-Based Archival Storage. [Online]. Available: http://www.techrepublic.com/whitepapers/pergamum-replacing-tape-with-energy-efficient-reliable-disk-based-archival-storage/1294097, [Accessed 2011-06-17].

[2] H. Newman, “LBL HPSS Workshop for DOE/SC,” CTO/Instrumental, Inc., Jul. 2009.

[3] J. Mahendra, (1998) What is archive? Definition from Whatis.com [Online]. Available: http://searchstorage.techtarget.com/definition/archive, [Accessed 2011-06-17].

[4] Wikipedia.org, (2011) Transaction processing system [Online]. Available: http://en.wikipedia.org/wiki/Transaction_processing_system, [Accessed 2011-06-17].

[5] H. Weber, Deutscher Wetterdienst (DWD), HPSS Reference and Booth Demo at official presentation at CeBIT 2011 trade fair, Hannover, Germany 2011.

[6] IBM Systems and Technology, (2010), Optimize storage capacity with IBM Real-time Compression, [Online]. Available: ftp://public.dhe.ibm.com/common/ssi/ecm/en/tsb03020usen/TSB03020USEN.PDF, [Accessed 2011-06-18].

[7] Bundesministerium für Bildung und Forschung (BMBF), (2009), Ergebnisse der ersten HPC- Fördermaßnahme, „HPC-Software für skalierbare Parallelrechner“, Bundesministerium für Bildung und Forschung, [Online]. Available: http://www.pt-it.pt-dlr.de/_media/HPC_Infoblatt.pdf, [Accessed 2011-06-23].

[8] wiseGEEK, (2003-2011), What is Data Transfer Rate? [Online]. Available: http://www.wisegeek.com/what-is-data-transfer-rate.htm, [Accessed 2011-06-23].

[9] H. Hulen and G. Jaquette, “Operational concepts and methods for using RAIT in high availability tape archives”, [Online]. Available: http://www.storageconference.org/2011/Papers/MSST.Hulen.pdf, [Accessed 2011-06-08].

[10] HPSS Collaboration.org, (2010), What is High Performance Storage System? [Online]. Available: http://www.hpss-collaboration.org/index.shtml, [Accessed 2011-06-18].

[11] National Energy Research Scientific Computing (NERSC) Facility, Extreme Scale Workshop, (2009), HPSS in the Extreme Scale Era, [Online]. Available: http://www.nersc.gov/assets/HPC-Requirements-for-Science/HPSSExtremeScaleFINALpublic.pdf, [Accessed 2011-06-18].

[12] H. Hulen, Basics of the High Performance Storage System, [Online]. Available: http://www.isi.edu/~annc/classes/grid/papers/HPSS-Basics.pdf, [Accessed 2011-06-18].

[13] Deutsches Klimarechenzentrum (DKRZ), (2011), Data Archive, [Online]. Available: http://www.dkrz.de/ Klimarechner-en/datenarchiv, [Accessed 2011-06-18].

[14] J. A. Gerry, H. Hulen, P. Schaefer, and B. Coyne, (2009), High Performance Storage System Overview, Slide Number 10, [Online]. Available: http://www.hpss-collaboration. org/documents/HPSSIntroduction2009.pdf, [Accessed 2011-06-18].

[15] Logic Supply, (2011) Transcend Commercial 2.5" SATA SSD, 64 GB , [Online]. Available: http://www.logicsupply. com/products/64gssd25s_m, [Accessed 2011-07-03].

[16] H. Hulen, personal communication per email, Jun. 17, 2011.

[17] S. Richards and F. Dequenne, "HPSS at ECMWF", High Performance Storage System User Forum (HUF) 2010 presentation, Sept. 28, 2010, Hamburg, Germany.

[18] M. Francis, UK Met Office, "HPSS User Forum", High Performance Storage User Forum (HUF) 2010 presentation, Sept. 28, 2010, Hamburg, Germany.

[19] University Corporation for Atmospheric Research (UCAR), (2011), CISL HPSS, [Online]. Available: http://www2. cisl.ucar.edu/book/export/html/964, [Accessed 2011-06-18].

[20] HPSS Collaboration.org, (2010), Learn about HPSS, The HPSS Collaboration, [Online]. Available: http://www.hpss-collaboration.org/hpss_collaboration.shtml, [Accessed 2011-06-18].

[21] IBM Systems and Technology, (2011), FAQ: IBM Real-time Compression [Online]. Available: http://www.realtime compression.com/library_brochures.asp, [Accessed 2011-06-18].

[22] Halliburton, (2009, Feb.), Press Release, Landmark introduces its PetroStor™ Cost-Competitive, Disk-Based tape Replacement Storage Solution for Oil and Gas Environments, [Online]. Available: http://www.realtimecompression.com/ content/press_releases/PetroStor_Storage_Solution_Press_Release.pdf, [Accessed 2011-06-26].

[23] D. Watson, (2007, Oct.), Yes, Virginia, There is an HPSS in Your Future, [Online]. Available: http://www.hpss-collaboration.org/documents/WatsonSalishan2007.pdf, [Accessed 2011-06-26].

[24] D. Boomer, K. Broussard, and M. Meseke, (2011), HPSS 8 Metadata Services Evolution to Meet the Demands of Extreme Scale Computing, [Online]. Available: http://www.pdsi-scidac.org/events/PDSW10/ resources/posters/HPSS8.pdf, [Accessed 2011-06-26].

High performance cosmological simulations on a grid of supercomputers

Derek Groen
Centre for Computational Science
University College London
London, United Kingdom
Email: d.groen@ucl.ac.uk

Steven Rieder, Simon Portegies Zwart
Leiden Observatory
Leiden University
Leiden, the Netherlands
Email: rieder@strw.leidenuniv.nl,
spz@strw.leidenuniv.nl

***Abstract*—We present results from our cosmological N-body simulation which consisted of 2048x2048x2048 particles and ran distributed across three supercomputers throughout Europe. The run, which was performed as the concluding phase of the Gravitational Billion Body Problem DEISA project, integrated a 30 Mpc box of dark matter using an optimized Tree/Particle Mesh N-body integrator. We ran the simulation up to the present day (z=0), and obtained an efficiency of about 0.93 over 2048 cores compared to a single supercomputer run. In addition, we share our experiences on using multiple supercomputers for high performance computing and provide several recommendations for future projects.**

***Keywords*-high-performance computing; distributed computing; parallelization of applications; cosmology; N-body simulation**

I. Introduction

Cosmological simulations are an efficient method to gain understanding of the formation of large-scale structures in the Universe. Large simulations were previously applied to model the evolution of dark matter in the Universe [1], and to investigate the properties of Milky-Way sized dark matter halos [2], [3]. However, these simulations are computationally demanding, and are best run on large production infrastructures. We have previously run a cosmological simulation using two supercomputers across the globe [4] with the GreeM integrator [5], [6], and presented the SUSHI N-body integrator [7], which we used to run simulations across up to four supercomputers. The simulations we ran in the Gravitational Billion Body Project produced over 110 TB of data, which we have used to characterize the properties of ultra-faint dwarf galaxies [8], and to compare the halo mass function in our runs to analytical formulae for the mass function. Among other things, we found that the halo mass function in our runs shows good agreement with the Sheth and Tormen function [9] down to $\sim 10^7$ solar mass.

Here we present the performance results of a production simulation across three supercomputers, as well as several other runs which all use an enhanced version of SUSHI. The production simulation ran continuously for ~ 8 hours, using 2048 cores in total for calculations as well as 4 additional cores for communications. We achieved a peak performance of 3.31×10^{11} tree force interactions per second, a sustained performance of 2.19×10^{11} tree force interactions per second and a wide area communication overhead of less than 10% overall.

We briefly reflect on the improvements made to SUSHI for this work in Section 2, while we report on tests performed on a single supercomputer in Section 3. In Section 4 we describe our experiments across three supercomputers and present our performance results. We reflect on our experiences on using multiple supercomputers for distributed supercomputing simulations, and provide several recommendations for users and resource providers in Section 4 and present our conclusions in Section 5.

A. Related work

There are a several other projects which have run high performance computing applications across multiple supercomputers. These include simulations of a galaxy collision [10], a materials science problem [11] as well as an analysis application for arthropod evolution [12]. A larger number of groups performed distributed computing across sites of PCs rather than supercomputers (e.g., [13], [14], [15]). Several software tools have been developed to facilitate high performance computing across sites of PCs (e.g., [16], [17], [18], [19], [20]) and within volatile computing environments [21]. The recently launched MAPPER EU-FP7 project [22] seeks to run multiscale applications across a distributed supercomputing environment, where individual subcodes periodically exchange information and (in some cases) run concurrently on different supercomputing architectures.

II. Improvements to SUSHI

Based on results of our earlier simulations and in preparation for the production run across three supercomputers we made several modifications to the SUSHI distributed N-body integrator. In our previous experiments a relatively large amount of computation and communication time was spent on (non-parallelize) particle-mesh integration. To reduce this bottleneck we now parallelized the particle-mesh integration routines using the parallel FFTW2 library [23] and a one dimensional slab decomposition. We also optimized the communications of the particle-mesh integration by introducing a scheme where sites only broadcast those mesh cells which have actual particle content. This

ISBN: 978-1-61208-161-8

Table I
INITIAL CONDITION AND ACCURACY PARAMETERS USED FOR OUR SIMULATIONS WITH 2048^3 PARTICLES.

Parameter	Value
Matter density parameter (ω_0)	0.3
Cosmological constant (λ_0)	0.7
Hubble constant (H_0)	70.0 km/s/Mpc
Mass fluctuation parameter (σ_8)	0.8
Box size	$(30\text{Mpc})^3$
Softening for 2048^3 particle run.	175 pc
Sampling rate for 2048^3 particle run.	20000

optimization reduced the size of the mesh communications by a factor roughly equal to the number of sites used, in the case of an equal domain distribution.

In some of the larger previous runs we also observed load imbalances if the code was run across two machines with different architectures, despite the presence of a load balancing scheme. This result has led us to further optimized the load balancing in SUSHI, taking into account not only the force integration time, but also the number of particles stored on each node. In addition to these changes, we also seized the opportunity to plug in a more recent MPWide [24] version into SUSHI. This newer version contains several optimizations to improve the wide area communication over networks with a high latency.

III. TESTS ON A SINGLE SITE

A. Setup

We performed a number of runs on the Huygens supercomputer to validate the scalability of our new implementation, and to provide performance measurements against which we can compare our results using multiple sites. More information on the Huygens machine can be found in the second column of Tab. III. The initial conditions for this simulation is the snapshot at redshift $z = 0.0026$ from the CosmoGrid simulation (described in [4]). We also use the simulation parameters chosen for the CosmoGrid simulation, which are summarized in Tab. I. Here the first four parameters are constants which are derived from WMAP observations (with a slight-roundoff) and the physical size of our simulated system is given by the fifth parameter (Box size). The softening in our simulation (i.e. a length value added to reduce the intensity of close interactions) and the sampling rate are given by the last two parameters. The sampling rate is the ratio of particles in the simulation divided by the number of particles sampled by the load balancing scheme. Our simulation used a mesh size of 512^3 cells. We ran the simulation using respectively 512 cores and 1024 cores until $z = 0.0024$, and using 2048 cores until the simulation completed (at $z = 0$). The number of force calculations per step in the simulation varies for different z values, though these variations are neglishible for $z < 0.01$.

Table II
OVERVIEW OF EXPERIMENTS PERFORMED WITH THE ENHANCED SUSHI CODE ON THE HUYGENS SUPERCOMPUTER. THE TIME SPENT ON COMMUNICATION IS GIVEN IN THE FOURTH COLUMN, WHILE THE TOTAL RUNTIME IS GIVEN IN THE FIFTH COLUMN. ALL TIMES ARE MEASURED PER STEP, AVERAGED OVER STEPS 1-11. IN ADDITION WE INCLUDED THE TIMING RESULTS OF THE LAST 10 STEPS OF THE SIMULATION RUNNING ON 2048 CORES (BOTTOM ROW).

N	p	θ	comm. t [s]	runtime [s]	z range $\times 10^{-3}$	speedup
2048^3	512	0.5	19.18	501.3	2.5-2.4	1
2048^3	1024	0.5	13.96	258.2	2.5-2.4	1.94
2048^3	2048	0.5	22.34	151.0	2.5-2.4	3.32
2048^3	2048	0.5	16.22	143.7	0.1-0.0	-

B. Results

The performance results of our runs are shown in Tab. II. In addition, the total runtime of the run using 2048 cores is given by the light blue line in Fig. 2. The overall performance of the code is dominated by calculations, with the communication overhead ranging from ~5% for 512 cores to ~10-15% for 2048 cores. During the run using 2048 cores, several snapshots were written. This resulted in a greatly increased execution time during two steps of the run.

IV. TESTS ACROSS THREE SITES

A. Setup

We performed our main run using a total of 2048 cores across three supercomputers, which are listed in Tab. III. These machines include Huygens in the Netherlands (1024 cores), Louhi in Finland (512 cores), and HECToR in Scotland (512 cores). The sites are connected to the DEISA shared network with either a 1Gbps interface (HECToR) or a 10Gbps interface (Huygens, Louhi). The initial conditions and simulation parameters chosen are identical to those of the runs using 1 supercomputer, although we use a mesh of 256^3 cells. The use of a smaller mesh size results in a slightly higher calculation time as tree interactions are calculated over a longer range, but a somewhat lower time spent on intra-site communications. We configured MPWide to use 64 parallel `tcp` streams per path for the wide area communication channels, each with a `tcp` buffer size set at 768 kB and packet-pacing set at 10 MB/s maximum. We enabled some load balancing during the run, though we had to limit the boundary moving length per step to 0.00001 of the box length due to memory constraints on our communication nodes and the presence of dense halos in our initial condition.

In addition to the main run, we also performed three smaller runs using the same code across the same three supercomputers. These include one run with 1024^3 particles using 80 cores per supercomputer, and two runs with 512^3 particles using 40 cores per supercomputer. These runs also used a mesh size of 256^3, though we did reduce the

Table III
PROPERTIES OF THE THREE SUPERCOMPUTERS USED FOR OUR RUN. THE MEASURED PEAK NUMBER OF TREE FORCE INTERACTIONS (IN MILLIONS) PER SECOND PER CORE IS GIVEN FOR EACH SITE IN THE BOTTOM ROW.

Name	Huygens	Louhi	HECToR
Location	Amsterdam	Espoo	Edinburgh
Vendor	IBM	Cray	Cray
Architecture	Power6	XT4	XT4
# of cores	3328	4048	12288
CPU [GHz]	4.7	2.3	2.3
RAM / core [GB]	4/8	1/2	2
force calcs. / core [Mints/s]	185	256	250

Table IV
OVERVIEW OF EXPERIMENTS PERFORMED WITH THE ENHANCED SUSHI CODE ACROSS ALL THREE SUPERCOMPUTERS. ALL TIMES ARE MEASURED PER STEP, AVERAGED OVER 10 STEPS.

N	p	θ	comm. time		runtime	z range
			WAN	total		
512^3	120	0.3	6.925	7.312	39.70	11.8-10.1
512^3	120	0.5	5.982	6.335	24.60	9.9-8.8
1024^3	240	0.3	12.09	14.04	214.5	17.0-14.9
2048^3	2048	0.5	15.40	24.77	167.7	0.0026-0.0025
2048^3	2048	0.5	14.62	23.13	155.2	0.0001-0

sampling rate to respectively 10000 and 5000 for the runs with 1024^3 and 512^3 particles. The force softening used for these runs were respectively 1.25kpc and 2.5kpc, and we set the boundary moving length limit to 0.01 of the box length. Some of the measurements were made using an opening angle θ of 0.3, rather than 0.5. Using a smaller opening angle results a higher accuracy of the force integration on close range, but also results in a higher force calculation and tree structure communication time per step.

B. Results

The timing results of our production run are shown in Fig. 1. Here, we also added the wall-clock time results of the simulation run using 2048 cores on Huygens as reference. The simulation run across three sites is only $\sim 9\%$ slower per step than the single-site run, despite the slightly higher force calculation time due to the lower number of mesh cells. The peaks in wall-clock time of the single site run are caused by the writing of snapshots during those steps (we only wrote one snapshot at the end of the three site run). The total wide area communication overhead of our run is $\lesssim 10\%$ at about $15s$ per step. Most of this time is required to exchange the tree structures between sites, though the communications for the parallelized particle-mesh require an additional $\sim 2.5s$ per step. Despite the use of a shared wide area network, the communication performance of our run shows very little jitter and no large slowdowns. We provide a snapshot of the final state of the simulation (at $z = 0$), distributed across the three supercomputers, in Fig. 2.

We also provide a numerical overview of the production run performance, as well as that of several other runs which use the new code, in Tab.IV. The communication overhead for the runs with 512^3 particles is less than 20%, while the overhead for the run with 1024^3 particles is just 6.5%. The parallelization of the particle-mesh integration and the enhanced load balancing greatly improved the performance of these runs, especially in the case with 1024^3 particles. Here, the communication overhead was reduced by $\sim 60\%$ and the overall runtime by more than 25% compared to the previous version [7].

V. USER EXPERIENCES

We have presented results from several cosmological simulations which run across three supercomputers, including a production run lasting for 8 hours. In the process of seeking a solution for wide area message passing between supercomputers, requesting allocations, arranging network paths and preparing for the execution of these simulations, we have learned a number of valuable lessons.

Primarily, we found that it is structurally possible to do high performace computing across multiple supercomputers. During the GBBP project we have run a considerable number of large-scale simulations using two or more supercomputers, with results improving as we were able to further enhance the N-body integrator and optimize the MPWide communication library for the wide area networks that we used.

The cooperation of the resource providers was particularly crucial in this project, as they enabled previously unavailable network paths and provided us with means to initiate simulations concurrently at the different sites. However, reserving networks and orchestrating concurrent supercomputer runs currently does require a disproportionate amount of time and effort, which makes performance optimization and debugging a challenging task. The effort required to run applications across supercomputers can be greatly reduced if resource providers were to adopt automated resource reservation systems for their supercomputers, and maintain shared high-bandwidth networking between sites. The persistent DEISA shared network connections helped greatly in our case, as we could use it at will without prior network reservations.

The software environment across different supercomputers, even within the same distributed infrastructure, is very heterogeneous. This made it unattractive to use existing middleware or message passing implementations to make different sites interoperable. We chose to use a modular approach where we connected platform-specific optimized versions of the SUSHI code with the MPWide communication library. With MPWide being a user-space tool that requires no external libraries or administrative privileges, we are able to install and run the simulation code in the locally preferred software environments on each site without needing any additional (grid) middleware. We recommend adopting a similar modular software approach in future distributed supercomputing efforts for its ease of installation and optimization, at least until resource providers present a homogeneous and interoperable software environment for distributed supercomputing.

This paper focuses on the calculation and communication performance aspects of a single application run across supercomputers. However, the methods presented here can be applied for several other purposes. During this project we were confronted with additional overhead introduced by disk I/O, as can be observed in Figure 1. With supercomputer disk performance and capacity improving at a much slower rate than the compute power, the deployment of an application across sites may help to eliminate a disk I/O performance bottleneck, though a detailed investigation will be needed to quantify such potential benefit. Additionally, the communication technique could be used to facilitate periodic exchanges between different simulation codes, each of which runs on a different site and tackles a different aspect of a complex multiscale or multiphysics problem.

VI. Conclusion

Our results show that cosmological production simulations run efficiently across supercomputers for a prolonged time. The political effort required to arrange cross-supercomputer runs is considerable, and is an important reason why few people have attempted to run production simulations across supercomputers. We have shown that the added overhead of using a network of supercomputers is rather marginal for at least one optimized production application and that given the right (political) environment, supercomputers can be conveniently connected to form even larger high performance computing resources.

Acknowledgements

We are grateful to Jeroen Bédorf, Juha Fagerholm, Tomoaki Ishiyama, Esko Keränen, Walter Lioen, Jun Makino, Petri Nikunen, Gavin Pringle and Joni Virtanen for their contributions to this work. This research is supported by the Netherlands organization for Scientific research (NWO) grant #639.073.803, #643.200.503 and #643.000.803, the Stichting Nationale Computerfaciliteiten (project #SH-095-08) and the MAPPER EU-FP7 project (grant no. RI-261507). We thank the DEISA Consortium (EU FP6 project RI-031513 and FP7 project RI-222919) for support within the DEISA Extreme Computing Initiative (GBBP project).

References

[1] V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman, A. Evrard, J. Colberg, and F. Pearce, "Simulations of the formation, evolution and clustering of galaxies and quasars," *Nature*, vol. 435, pp. 629–636, Jun. 2005.

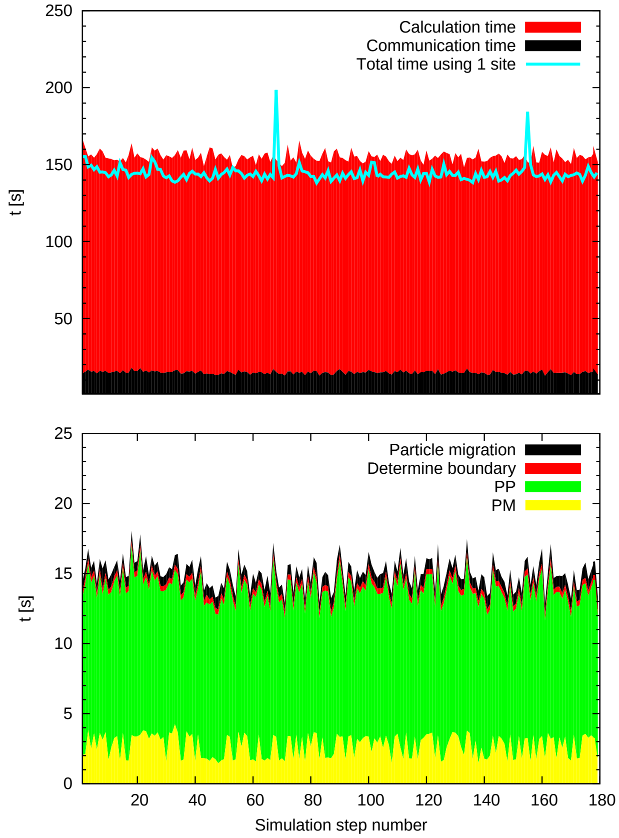


Figure 1. Performance results of the production simulation across three sites. In the top figure we provide the total time spent on calculation per step in red, and on communication per step in blue. Here, the total wall-clock time of an identical simulation using 2048 processes only on Huygens is given by the light blue line. Time spent on the four communication phases is given in the bottom figure. These phases are (from top to bottom) the migration of particles between sites, the exchanges of sample particles for determining the site boundaries, the local essential tree exchanges (PP) and the mesh cell exchanges (PM). See [7] for full details on the communication routines of the code.

[2] V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. M. White, "The Aquarius Project: the subhaloes of galactic haloes," *MNRAS*, vol. 391, pp. 1685–1711, Dec. 2008.

[3] T. Ishiyama, T. Fukushige, and J. Makino, "Variation of the Subhalo Abundance in Dark Matter Halos," *ApJ*, vol. 696, pp. 2115–2125, May 2009.

[4] S. Portegies Zwart, T. Ishiyama, D. Groen, K. Nitadori, J. Makino, C. de Laat, S. McMillan, K. Hiraki, S. Harfst, and P. Grosso, "Simulating the universe on an intercontinental grid," *Computer*, vol. 43, pp. 63–70, 2010.

[5] K. Yoshikawa and T. Fukushige, "PPPM and TreePM Methods on GRAPE Systems for Cosmological N-Body Sim-

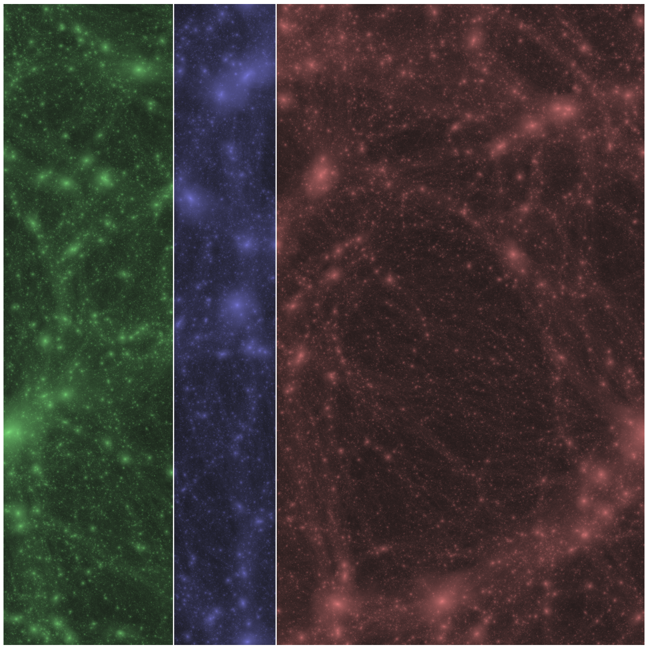


Figure 2. The final snapshot of the production simulation across three supercomputers, taken at $z = 0$. The size of the box is 30x30x30 Mpc and the contents are colored to match the particles residing in Espoo (green, left), Edinburgh (blue, center) and Amsterdam (red, right) respectively.

ulations," *Publications of the Astronomical Society of Japan*, vol. 57, pp. 849–860, Dec. 2005.

[6] T. Ishiyama, T. Fukushige, and J. Makino, "GreeM: Massively Parallel TreePM Code for Large Cosmological N -body Simulations," *Publications of the Astronomical Society of Japan*, vol. 61, pp. 1319–1330, Dec. 2009.

[7] D. Groen, S. Portegies Zwart, T. Ishiyama, and J. Makino, "High Performance Gravitational N-body simulations on a Planet-wide Distributed Supercomputer," *Computational Science and Discovery*, vol. 4, no. 015001, Jan. 2011.

[8] T. Ishiyama, J. Makino, S. Portegies Zwart, D. Groen, K. Nitadori, S. Rieder, C. de Laat, S. McMillan, K. Hiraki, and S. Harfst, "The Cosmogrid Simulation: Statistical Properties of Small Dark Matter Halos," *ArXiv e-prints (submitted to PASJ)*, Jan. 2011.

[9] R. K. Sheth and G. Tormen, "Large-scale bias and the peak background split," *MNRAS*, vol. 308, pp. 119–126, Sep. 1999.

[10] M. Norman, P. Beckman, G. Bryan, J. Dubinski, D. Gannon, L. Hernquist, K. Keahey, J. Ostriker, J. Shalf, J. Welling, and S. Yang, "Galaxies collide on the i-way: an example of heterogeneous wide-area collaborative supercomputing," *International Journal of High Performance Computing Applications*, vol. 10, no. 2-3, pp. 132–144, 1996.

[11] T. J. Pratt, L. G. Martinez, M. O. Vahle, and T. V. Archuleta, "Sandia‘s network for supercomputer ‘96: Linking supercomputers in a wide area asynchronous transfer mode (atm) network," Sandia National Labs., Albuquerque, NM (United States), Tech. Rep., 1997.

[12] C. Stewart, R. Keller, R. Repasky, M. Hess, D. Hart, M. Muller, R. Sheppard, U. Wossner, M. Aumuller, H. Li, D. Berry, and J. Colbourne, "A global grid for analysis of arthropod evolution," in *GRID '04: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 328–337.

[13] A. Gualandris, S. Portegies Zwart, and A. Tirado-Ramos, "Performance analysis of direct n-body algorithms for astrophysical simulations on distributed systems." *Parallel Computing*, vol. 33, no. 3, pp. 159–173, 2007.

[14] H. Bal and K. Verstoep, "Large-scale parallel computing on grids," *Electronic Notes in Theoretical Computer Science*, vol. 220, no. 2, pp. 3 – 17, 2008, proceedings of the 7th International Workshop on Parallel and Distributed Methods in verifiCation (PDMC 2008).

[15] P. Bar, C. Coti, D. Groen, T. Herault, V. Kravtsov, M. Swain, and A. Schuster, "Running parallel applications with topology-aware grid middleware," in *Fifth IEEE international conference on e-Science and Grid computing: Oxford, United Kingdom*. Piscataway, NJ: IEEE Computer Society, December 2009, pp. 292–299.

[16] N. Karonis, B. Toonen, and I. Foster, "Mpich-g2: A grid-enabled implementation of the message passing interface," *Journal of Parallel and Distributed Computing*, vol. 63, no. 5, pp. 551 – 563, 2003, special Issue on Computational Grids.

[17] E. Gabriel, M. Resch, T. Beisel, and R. Keller, "Distributed computing in a heterogeneous computing environment," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, ser. Lecture Notes in Computer Science, vol. 1497. Springer, 1998, pp. 180–187.

[18] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, "Open MPI: Goals, concept, and design of a next generation MPI implementation," in *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, September 2004, pp. 97–104.

[19] S. Manos, M. Mazzeo, O. Kenway, P. V. Coveney, N. T. Karonis, and B. R. Toonen, "Distributed mpi cross-site run performance using mpig," in *HPDC*, 2008, pp. 229–230.

[20] S. Sundari M., S. S. Vadhiyar, and R. S. Nanjundiah, "Morco: middleware framework for long-running multi-component applications on batch grids," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 328–331. [Online]. Available: http://doi.acm.org/10.1145/1851476.1851522

[21] B. Rood, N. Gnanasambandam, M. J. Lewis, and N. Sharma, "Toward high performance computing in unconventional computing environments," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: ACM, 2010, pp. 627–635. [Online]. Available: http://doi.acm.org/10.1145/1851476.1851569

[22] MAPPER, “Multiscale applications on european e-infrastructures: http://www.mapper-project.eu,” Jul. 2011.

[23] M. Frigo and S. Johnson, “The design and implementation of fftw3,” in *Proceedings of the IEEE*, vol. 93, Feb 2005, pp. 216–231.

[24] D. Groen, S. Rieder, P. Grosso, C. de Laat, and P. Portegies Zwart, “A light-weight communication library for distributed computing,” *Computational Science and Discovery*, vol. 3, no. 015002, Aug. 2010.

Mantle Convection in a 2D Spherical Shell

Ana-Catalina Plesa
Dept. of Planetary Physics,
Joint Planetary Interior Physics Research Group
University of Münster and IfP DLR Berlin
Berlin, Germany
e-mail: ana.plesa@dlr.de

***Abstract*—To get a closer look inside the planets and evaluate their mantle dynamics, numerical simulation of thermal convection has proved itself to be a powerful tool. In order to achieve a high resolution, to obtain faster results and to study a larger parameter range, the simulation of mantle convection in a two-dimensional spherical geometry is more efficient than a full three-dimensional spherical shell. In this work, we show the performance and results of a 2D version of GAIA, a mantle convection spherical code with strongly temperature dependent rheology.**

Keywords-mantle convection; 2D spherical code; performance; domain decomposition.

I. Introduction

Numerical simulations have been used to model mantle convection, which may take different forms depending on the planet. On Earth, mantle convection involves recycling of the surface or oceanic lithosphere and results in plate tectonics. Because the lithosphere is relatively cold, recycling the lithosphere represents an extremely efficient way to remove the heat and cool the mantle. On other terrestrial planets, the so-called one-plate planets like Venus and Mars, mantle convection does not involve the outer layers. Instead it occurs below a stagnant lid where heat is transported by conduction. The different characteristics of mantle convection have also a strong influence on the resources needed to simulate the interior dynamics of a planet.

Mantle convection is a highly non-linear process which can be modeled using the conservation equations of the mass, energy and momentum [15]. The simulation time depends on various factors. The size of the grid used for space discretization plays an important role but also the number of time steps needed to reach a solution of the conservation equations is crucial. For realistic calculations a high resolution and small time steps are needed for the convergence of the solution. The computational time increases considerably and thus 2D models are more suited to perform such resource demanding simulations. To run a simulation with a reasonable resolution, the code must work with more than one CPU in parallel.

In the next section several approaches for the domain decomposition of a 2D spherical grid are presented. We also introduce a formula to calculate the overhead of data exchange between the domains for the two- dimensional spherical grid. In Section III, we illustrates the speedup obtained for 2D grids with up to 128 CPUs on different supercomputer centers. Another performance test consists in the amount of time the simulation needs depending on the size of the grid to reach a stable (steady-state) solution. We present a comparison of the required computational time for both 2D and 3D grids. Section IV illustrates results obtained by using the GAIA framework with two and three-dimensional spherical grids. In order to compare these results, the Nusselt number and plots showing temperature distribution for each case have been computed and compared. We also compare our results obtained with the 2D version of the GAIA code to other published results. In the final section we present our conclusions and give an outlook on future works.

II. Domain Decomposition

The space discretization is based on the finite-volume method with the advantage of utilizing fully irregular grids in three [9], [10] and two dimensions [13]. The grid contains Voronoi cell information, obtained by computing the Voronoi diagram after performing the Delaunay triangulation of the computational points as shown in Figure 1. To run the code with more than one CPU in parallel a domain decomposition of the grid is applied, which results in an optimal breakdown of the grid into p equal surfaces, where p specifies the amount of domains and processors. An efficient domain decomposition minimizes the area between these sections, leading to a minimized overhead of data exchange between the processors.

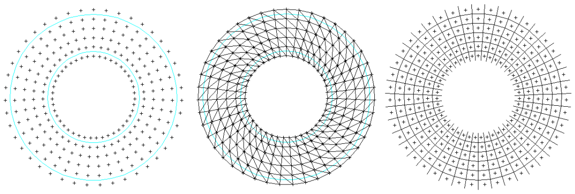


Figure 1. Left: grid with computational nodes, center: Delaunay triangulation of the grid points, right: Voronoi diagram of the 2D spherical grid.

Halo-cells, sometimes called ghost-cells, arise in domain decomposition as additional cells in each domain, which form an overlapping zone where data is exchanged. These cells border each domain and are on the same position as their active cells on the neighboring domain. The ratio between halo-cells and grid cells is a first measure of efficiency for parallelization. This ratio is important to determine the amount of data exchanged between the domains.

Using the GAIA Framework with a 2D spherical grid a circle surface will be decomposed into sectors with the same area by equally distributing p potential points on the circle circumference. Thus the coordinates of the i-th potential point can be calculated as follows:

$$p_i.x = \cos(2i\pi/p),\ p_i.y = \sin(2i\pi/p) \qquad (1)$$

The nodes within each sector is assigned to a single processor. The processor assigned to a sector is the one corresponding to the point on the circle circumference closest to the grid nodes in that sector. In Figure 2, we show three possible domain decomposition approaches with the resulting halo-zone: To achieve similar lateral and radial resolution, the number of points per shell is always greater than the number of shells. Keeping this in mind, the first approach for the domain decomposition, where the domains are divided only radial, is the one with the highest number of halo-cells, since in this case the amount of halo-cells scales with the number of points per shell times the number of domains. In the following, we will investigate the performance of the other two approaches (RadialSplit and LateralSplit).

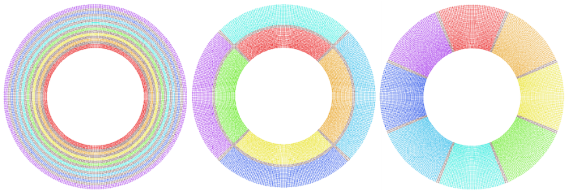


Figure 2. Domain decomposition: left: only radial, center: radial and lateral (RadialSplit), right: only lateral (LateralSplit); each color shows one domain and thus one processor; the gray regions show the Halo-zone between the domains.

III. Performance

The code was tested using four supercomputer centers: HLRN (North-German Supercomputing Alliance), PFCLUSTER1 (German Aerospace Centre, DLR Berlin), HP XC4000 (Steinbuch Centre for Computing, SCC Karlsruhe) and Itasca (Minnesota Supercomputing Institute for Advanced Computational Research). At HLRN, the HLRN-II SGI Altix ICE 8200 Plus cluster with computational nodes containing two quad-core sockets each for Intel Xeon Harpertown processors with 3 GHz and 16 GB memory per node has been used. The computational nodes on PFCLUSTER1 have each two quad-core AMD Opteron(tm) processors running at 2.3 GHz and 16 GB memory per node. On the XC2-Karlsruhe we tested the code with four-way computational nodes each containing two AMD Opteron Dual Cores running at 2.6 GHz with 32 GB per node. On the Itasca cluster we used computational nodes having each two quad-core Intel Xeon X5560 processors running at 2.8 GHz and 24 GB memory per node. In Figure 3, we show the speedup using a 128 shells grid with 152064 computational points which is a typical resolution for our mantle convection simulations. For the runs we used 8 CPUs, 16 CPUs, 32 CPUs, 64 CPUs and 128 CPUs in parallel.

To evaluate the performance, the same initial setup was tested on several node counts. The ratio of the execution time determines the speed-up. This speed-up is therefore the factor that determines the acceleration of the code for the same problem on various CPU counts. The speed-up has been calculated by averaging the time needed for a time-step over 20 time-steps. The ”speed-up” factor in Figure 3 is calculated by dividing the amount of time needed with eight CPUs by the amount of time needed with the parallel code. The dotted line in Figure 3 shows the optimal speed-up. Up to 64 CPUs the speedup increases. However, increasing the number of CPUs the number of halo-cells also increases and the performance will eventually drop due to communication overhead.

The speedup calculation in Figure 3 shows a better performance for the Intel Xeon Harpertown processors on HLRN. PFCLUSTER1, Itasca and XC2-Karlsruhe show similar speedup for 128 CPUs although on XC2-Karlsruhe the number of cores per node is limited to 4 while on all other clusters we used 8 cores per node. For 128 CPUs a slower increase in the performance can be observed due to the communication overhead. Figure 4 shows the performance obtained when using various grids and both only lateral divisions (LateralSplit) and lateral and radial

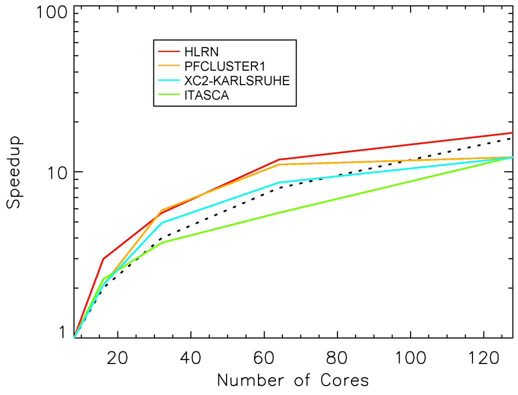


Figure 3. Speedup using a 2D 128 shells grid with 152064 computational nodes on 8, 16, 32, 64 and 128 CPUs on HLRN (red), PFCLUSTER1 (orange), XC2-Karlsruhe (blue) and Itasca (green).

divisions (RadialSplit) for the domain decomposition. The RadialSplit shows in most of the cases better performance than the LateralSplit.

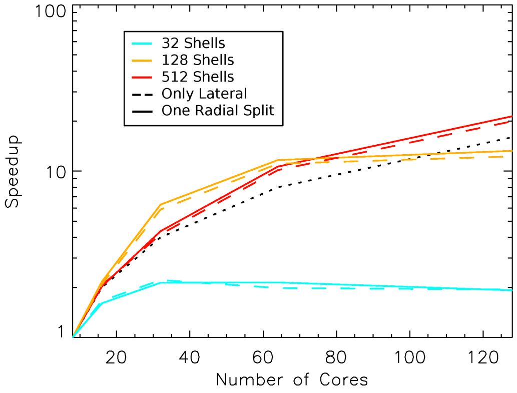


Figure 4. Speedup using three different 2D grids: 32 shells with 9056 computational points grid (blue), 128 shells with 152064 computational points grid (orange) and 512 shells with 2378752 grid (red) and both LateralSplit (dashed lines) and RadialSplit (full lines) domain decompositions.

For 2D grids one can use the following formula to calculate the number of halo-cells needed for both types of domain decomposition.

$$n_{cpus} = lS \cdot (rS + 1)$$
$$haloCells = 2 \cdot (ppS \cdot rS + s \cdot lS) \qquad (2)$$

where lS is the number of lateral domains, rS is the number of radial domains, ppS is the number of points per shell and s is the number of shells.

Using formula (2) we can calculate the number of halo-cells for a 2D grid knowing the the number of points per shell, the number of shells and the number of lateral and radial divisions. By increasing the grid's radial resolution and thus the number of shells or by increasing the number of CPUs used, the RadialSplit shows less halo-cells than the LateralSplit. Therefore the amount of exchanged data is smaller for larger grids and/or larger number of cores used.

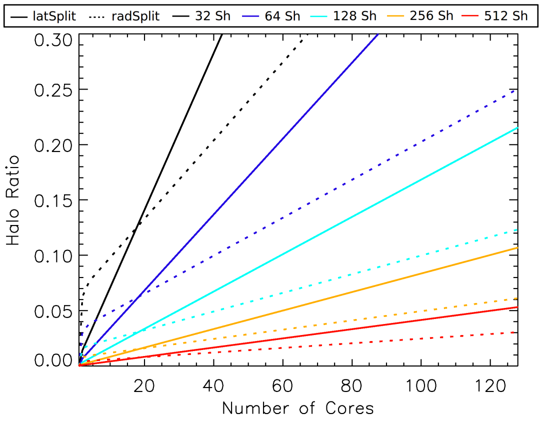


Figure 5. Ratio of halo-cells to computational cells depending on the grid resolution and the number of CPUs used.

In the next table we calculate the amount of time needed to reach a stable (steady-state) solution by using different resolutions for both two- and three-dimensional grids. The 2D grid is up to one order of magnitude faster than the three-dimensional grid with the same resolution. This is an major advantage when high resolution is needed or a larger parameter space has to be tested.

Table I
COMPUTATION TIME DEPENDING ON THE NUMBER OF GRID POINTS AND THE PROBLEM DIMENSION. WE USED THE ISOVISCOUS BENCHMARK-TEST FROM [9] AND 8 CPUS OF AN AMD OPTERON ARCHITECTURE.

Number of shells	Number of points (2D)	Computation time (s) (2D)	Number of points (3D)	Computation time (s) (3D)
16	3114	242.807	46116	2208.531
24	6760	399.519	266292	16797.99
32	11764	437.094	348228	30674.416
40	18186	836.693	430164	45135.616
48	25950	1355.142	512100	65438.916

IV. Application to Mantle Convection

A. Mantle Convection Model

We consider thermal convection in a 2D spherical shell using the GAIA code [9], [10]. The equations used are the equations of conservation of mass momentum and energy [15]. These equations are scaled with the thickness of the mantle as a length scale and with the thermal diffusivity as a time scale. Therefore the non-dimensional equations of a Boussinesq fluid assuming a Newtonian rheology and an infinite Prandtl number are [5]:

$$\nabla \cdot \vec{u} = 0 \qquad (3)$$
$$\nabla \cdot \left[\eta(\nabla\vec{u} + (\nabla\vec{u})^T)\right] + RaT\vec{e_r} - \nabla p = 0 \qquad (4)$$
$$\frac{\partial T}{\partial t} + \vec{u}\nabla T - \nabla^2 T - \frac{Ra_Q}{Ra} = 0 \qquad (5)$$

The parameters in the above and following equations are non-dimensionalized using the relationships to physical properties presented in [3] where $\vec{u}$ is the velocity field, η is the viscosity, T is the temperature, $\vec{e_r}$ is the unity vector in radial direction, p is the pressure, t is the time, Ra is the thermal Rayleigh number and Ra_Q is the Rayleigh number for internal heat sources.

The viscosity is calculated using the Arrhenius law for diffusion creep [11]. The non-dimensional formulation of the Arrhenius viscosity law for only temperature dependent viscosity [14] is given by:

$$\eta(T) = \exp\left(\frac{E}{T + T_{surf}} - \frac{E}{T_{ref} + T_{surf}}\right) \qquad (6)$$

where E is the activation energy, T_{surf} the surface temperature and T_{ref} the reference temperature.

We choose a fix surface temperature that will not change during the simulation. Depending on the problem, one can choose between no-slip and free slip boundary conditions. For this the velocity vector is decomposed into a lateral part projected onto the boundary and a radial part. In the free slip case the radial component of the velocity is set to zero while material can still move along the boundary whereas in the no-slip case both radial and lateral components are set to zero.

As mentioned earlier, the discretization of the governing equations is based on the finite-volume method with the advantage of utilizing fully irregular grids.

As space is discretized by a fixed grid, time must be discretized as well. For the temporal discretization a fully implicit second-order method, also called an implicit three-level scheme, as shown in [7] has been used. In contrast to spatial discretization, the temporal discretization is flexible and can adapt with a varying time step Δt to the situation. A method proposed by [2] and [12] called SIMPLE was adopted to solve the coupling of the continuity equation with the momentum equation.

Following quantities will are used for the comparison: the root mean square velocity, v_{rms}, and the volume averaged temperature $|T|$. Another value of interest is the Nusselt number, which is defined as the ratio of total heat flux to purely conductive heat-flux. Nu_{top} is the Nusselt number at the surface while Nu_{bottom} is the bottom Nusselt number. Nu_{avg} is the average between Nu_{top} and Nu_{bottom}.

B. 2D-3D Comparison

First we compare the results obtained using a 2D grid to the results obtained using a 3D grid. A spherical harmonics disturbance pattern has been added to the initial temperature field to force the convection to establish a certain symmetry. We choose here the cubic pattern from [10], a pattern which is widely used in steady-state benchmark tests.

When comparing 2D and 3D cases there are some differences which arise from the disagreement in the ratio between inner and outer surface of the 2D and 3D grid respectively [8]. In [17], a scaling was proposed. The inner and outer surface areas of the 2D grid can be fitted the area ratio in the 3D geometry. However this scaling has as a result a smaller inner radius which leads to crowding of structures near the inner portion of the 2D spherical grid [8]. For the comparison we choose an isoviscouse bottom-heated convection with a Rayleigh number of $1e4$ and free-slip boundary conditions for the velocity.

In Figure 6, we present the temperature distribution for both two- and three-dimensional cases. In Table II, we present output values obtained with both the 2D and 3D version of the GAIA code.

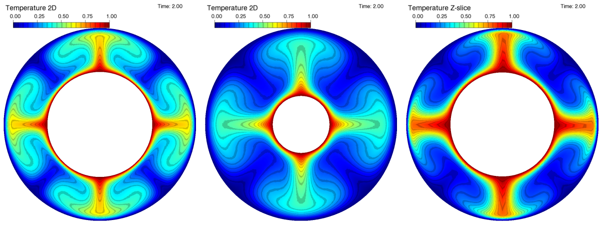


Figure 6. 2D-3D comparison for $Ra = 1e4$, $\Delta\eta_T = 1$: left 2D temperature distribution; center: 2D scaled radii temperature distribution; right: 3D temperature distribution.

Table II
COMPARISON OF THE RESULTS OBTAINED USING THE 2D VERSION AND 3D VERSION OF THE GAIA CODE; THE PARAMETERS USED ARE $Ra = 1e4$, $\Delta\eta_T = 1$ AND FREE SLIP BOUNDARIES.

Case	v_{rms}	$\|T\|$	Nu_{top}	Nu_{bot}	Nu_{avg}
GAIA 2D	46.239	0.382	4.710	4.662	4.686
GAIA 2D scaled	29.817	0.277	3.721	3.633	3.677
GAIA 3D	39.243	0.208	4.029	4.021	4.025

C. Comparison with COMSOL Multiphysics © 3.5

Next we show a comparison with the commercial product COMSOL Multiphysics © 3.5. The sets of tests used for comparison with the COMSOL software [4] include one isoviscouse test with $Ra = 1e4$ and one temperature-dependent viscosity test with $Ra_{0.5} = 1e4$ at a non-dimensional reference temperature $T_{ref} = 0.5$. The activation energy and surface temperature from equation (6) are chosen in such a way that the viscosity contrast across the computational domain is $\Delta\eta_T = 1e6$. For the tests computed using the GAIA code, we use a projected 2D grid, while in COMSOL the mesh is fully irregular. While with GAIA uses finite volume discretization, the discretization scheme in COMSOL is finite element based. In contrast to the 2D-3D comparison tests from the last subsection, we use here no slip top boundary condition for the momentum equation to suppress unrealistic zero-mode from appearing in COMSOL. In Figure 7, we present the results for the isoviscouse case. Both the temperature slice and the velocity field indicate a good agreement between the two codes. The temperature distribution show the same convection structure with fout thermal upwellings (plumes) on the axes. The maximum velocity is 49.54 for the GAIA case and 49.495 for the COMSOL run and exhibits identical distribution with high velocity in the areas where a thermal upwelling or a downwelling forms. The next comparison shows a temperature-dependent viscosity case using the Arrhenius viscosity law. In Figure 8, the temperature distribution show similar structure to the previous isoviscouse cases, however due to the temperature-dependent viscosity, the upwelling form changes since the bouyancy term in the conservation of momentum equation decreases with increasing viscosity.

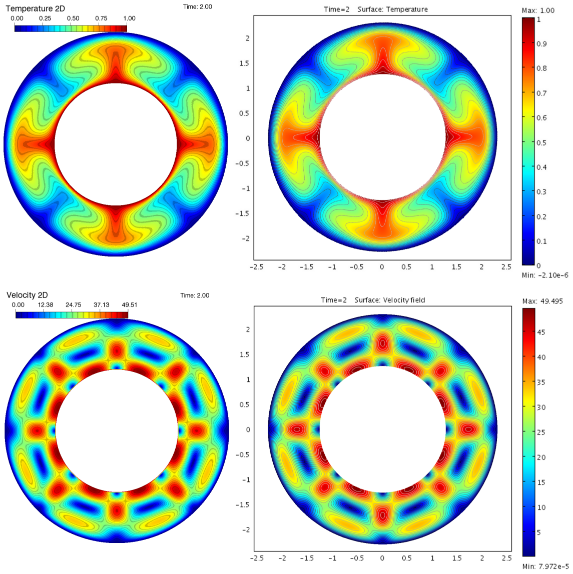


Figure 7. 2D comparison for $Ra = 1e4$, $\Delta\eta_T = 1$; left: 2D temperature (top) and velocity (bottom) distribution with GAIA; right: 2D temperature (top) and velocity (bottom) distribution with COMSOL Multiphysics © 3.5.

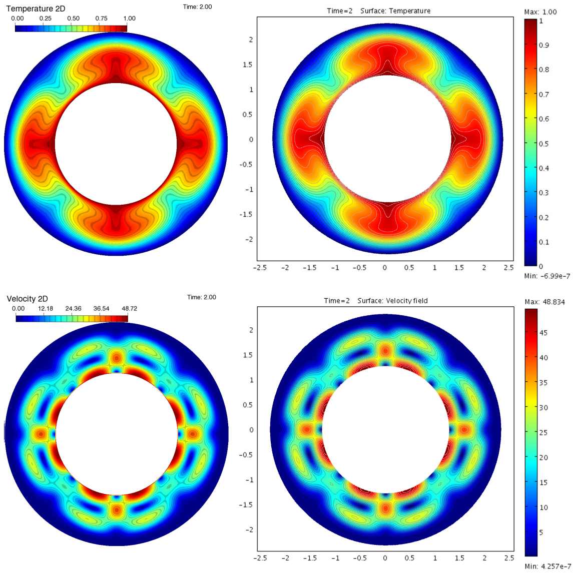


Figure 8. 2D comparison for $Ra_{0.5} = 1e4$, $\Delta\eta_T = 1e6$; left: 2D temperature (top) and velocity (bottom) distribution with GAIA; right: 2D temperature (top) and velocity (bottom) distribution with COMSOL Multiphysics © 3.5.

The highest velocity is in the regions of high temperature (thermal upwellings) since there the viscouse forces are weaker than in the rest of the mantle. In these cases, due to the temperature-dependent viscosity, a stagnant lid forms on top of the convecting mantle.

In Table III, we show further output values for both the isoviscouse cases ($\Delta\eta_T = 1$) and the temperature-dependent viscosity case ($\Delta\eta_T = 1e6$) obtained with the 2D Version of the GAIA code and with the COMSOL Multiphysics © 3.5 software.

Table III
COMPARISON OF THE RESULTS OBTAINED USING THE GAIA CODE AND COMSOL MULTIPHYSICS © 3.5 SOFTWARE; THE PARAMETERS USED ARE $Ra = 1e4$, AND NO-SLIP BOUNDARIES.

Case	v_{rms}	$\lvert T \rvert$	Nu_{top}	Nu_{bot}	Nu_{avg}
GAIA 2D $\Delta\eta_T = 1$	27.347	0.488	3.255	3.222	3.2385
COMSOL 2D $\Delta\eta_T = 1$	24.966	0.488	3.238	3.139	3.1885
GAIA 2D $\Delta\eta_T = 1e6$	19.396	0.535	2.143	2.121	2.132
COMSOL 2D $\Delta\eta_T = 1e6$	15.257	0.535	2.124	2.087	2.1055

D. Comparison with published results

Next we compare 2D results with other published results. The following table lists benchmark results for an isoviscouse case and a temperature-dependent viscosity case (Bl stands for [1] and Ha for [6]). Ra_1 is the bottom Rayleigh number. In Figure 9 and Figure 10 we show the temperature distribution and the Nusselt number. The results listed in

Table IV
COMPARISON WITH PUBLISHED RESULTS; BL STANDS FOR BLANKENBACH ET AL.[1] AND HA STANDS FOR HANSEN ET AL.[6].

Case	Grid	Ra	$\Delta\eta_T$	Nu_{top}	Nu_{bot}	Nu_{avg}
Bl	18x18	1e7	1e3	–	–	9.57
	24x24	1e7	1e3	–	–	9.63
This	32x325	1e7	1e3	9.5	9.6	9.55
Ha	60x180	1e6	1	–	–	17.20
This	96x1017	1e6	1	16.83	17.17	17.0

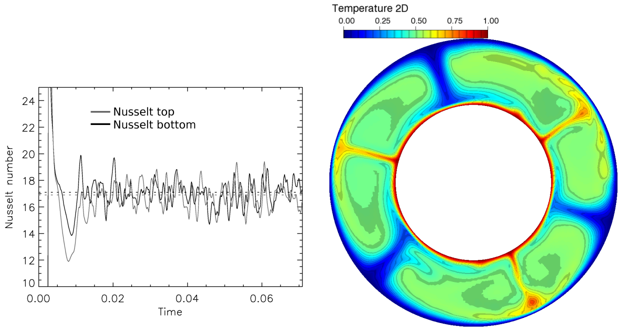


Figure 9. Left: Nusselt numbers as a function of time and right: 2D Temperature slice corresponding to the case from Blankenbach et al. [1].

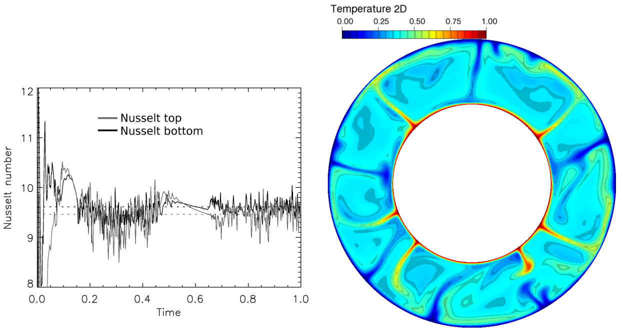


Figure 10. Left: Nusselt numbers as a function of time and right: 2D Temperature slice corresponding to the case from Hansen et al. [6].

Table IV show a good agreement between the two cases computed using GAIA 2D framework and the corresponding cases from [1] and [6]. Although the models from [1] and [6] use 2D box grids, the differences in the Nusselt numbers between [1], [6] and GAIA are within 1.2%. The two cases presented in Figure 10 and Figure 11 show a time-dependent behavior. Unlike the previous cases, the Nusselt number plotted in the right part of the figure varies with time until a quasi-steady-state is reached where the Nusselt number oscillates around a mean value.

V. Conclusion and Future Works

In this paper, we presented several methods for domain decomposition of 2D spherical grids and a formula for computing the resulting number of halo-cells and therefore the communication overhead. The performance tests show a super-linear speedup for the 2D simulations and a computation time needed to reach a steady-state solution at least one order of magnitude smaller than the one needed for the three dimensional cases. Further, comparison of 2D and 3D results show a good agreement in convection mode (number of thermal upwellings and downwellings) and Nusselt number. The validation with the commercial product COMSOL Multiphysics © 3.5 yieled satisfying results for both isoviscouse and temperature-dependent viscosity cases. Comparison with other published results showed similar results for the Nusselt numbers.

A future goal is to include active compositional fields in our model. For this, further benchmarks as well as performance tests will be needed.

Acknowledgment

This research has been supported by the Helmholtz Association through the research alliance ”Planetary Evolution and Life”. The author thanks the anonymous reviewers for their comments, which helped improving the manuscript.

References

[1] B. Blankenbach et al., *A benchmark comparison for mantle convection codes*, Geophys. J. Int., 1989, 98, 23-38.

[2] L.S. Caretto, A.D. Gosman, S.V. Patankar, and D.B. Spalding, *Two calculation procedures for steady, three-dimensional flows with recirculation*, Third Int. Conf. Numer. Methods Fluid Dyn., Paris, 1972.

[3] U. Christensen, *Convection with pressure- and temperaturedependent non-Newtonian rheology*, Geophysical Journal- Royal Astronomical Society, 1984, 77, 343-384.

[4] G., Couberbaisse, *A comparison between the software COMSOL © and a lattice Boltzmann code when simulating behaviour of viscous material*, European COST P19 project, CONFERENCE on Multiscale Modelling of Materials, Brno, Academy of Sciences - Masaryk University, Czech Republic.

[5] O. Grasset and E.M. Parmentier, *Thermal convection in a volumetrically heated, infinite Prandtl number fluid with strongly temperature-dependent viscosity: implications for planetary thermal evolution*, J. geophys. Res., 1998, 103, 18 17118 181.

[6] U. Hansen, D.A. Yuen, and A.V. Malevsky, *Comparison of steady-state and strongly chaotic thermal convection at high Rayleigh number*, Physical Review A, 1992, 46, 4742-4754.

[7] H. Harder and U. Hansen, *A finite-volume solution method for thermal convection and dynamo problems in spherical shells*, Geophysical Journal International, 2005, 161, 522- 532.

[8] J. W. Hernlund and P. J. Tackley, *Modeling mantle convection in the spherical annulus*, Phys. Earth Planet. Inter., 171 (1-4), 48-54. doi: 10.1016/j.pepi.2008.07.037.

[9] C. Huettig and K. Stemmer, *The spiral grid: A new approach to discretize the sphere and its application to mantle convection*, Geochem. Geophys. Geosyst., 9, Q02018, 2008, doi:10.1029/2007GC001581.

[10] C. Huettig and K. Stemmer, *Finite volume discretization for dynamic viscosities on Voronoi grids*, Phys. Earth Planet. Interiors, 2008, doi: 10.1016/j.pepi.2008.07.007.

[11] S. I., Karato and P. Wu, *Rheology of the upper mantle: a synthesis*, Science 260, 771-778, 1986.

[12] S.V. Patankar, *Numerical heat transfer and fluid flow*, edn, Vol., pp Pages, Mc-Graw-Hill, New York, 1980.

[13] A.-C. Plesa and C. Huettig, *Numerical Simulation of Planetary Interiors: Mantle Convection in a 2D Spherical Shell*, (Abstract), Workshop on Geodynamics 2008, Herz- Jesu-Kloster, Neustadt, Waldstr. 145 67434, Neustadt/Weinstrasse, Sept 30 - Oct 2, 2008.

[14] J. H., Roberts and S., Zhong, *Degree-1 convection in the Martian Mantle and the origin of the hemispheric dichotomy*, Journal of Geophysical Research E: Planets 111, 2006.

[15] G. Schubert, D.L. Turcotte, and P. Olson, *Mantle Convection in the Earth and planets*, Cambridge University Press, 2001.

[16] V.S. Solomatov and L.-N. Moresi, *Scaling of time-dependent stagnant lid convection: Application to small-scale convection on the earth and other terrestrial planets*, J. Geophys. Res., 2000,105, 21795-21818.

[17] P.E., van Keken, *Cylindrical scaling for dynamical cooling models of the earth*, Phys. Earth Planet. Interiors, 2001, 124, 119-130.

Constructing Parallel Programs Based on Rule Generators

Kiyoshi Akama
Information Initiative Center
Hokkaido University
Hokkaido, Japan
Email: akama@iic.hokudai.ac.jp

Ekawit Nantajeewarawat
Computer Science Program
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand
Email: ekawit@siit.tu.ac.th

Hidekatsu Koike
Faculty of Social Information
Sapporo Gakuin University
Hokkaido, Japan
Email: koike@sgu.ac.jp

***Abstract*—We propose a new architecture of parallel programs based on the master-worker model of parallel computing. In this architecture, computation is realized by rule application and rule generation. A master has a set of equivalent transformation rules (ET rules) and solves a problem by successively applying ET rules to definite clauses representing its computation state. A worker has a rule generator and makes computation by generating ET rules on demand based on run-time content of the master's computation states. A general scheme for constructing parallel programs based on rule-set generators and rule-generator generators is presented and a sufficient condition for the correctness of the scheme is established. Application of our framework to solving a constraint satisfaction problem is illustrated.**

***Keywords*-parallel computation; program correctness; rule generation; equivalent transformation**

I. Introduction

Constructing a correct parallel program that makes effective use of computing resources in a distributed environment is a nontrivial task. Major difficulties include how to strictly ensure the correctness of computation and how to obtain substantial efficiency gains, in particular under situations when the response time of distributed processes varies and is often unpredictable. Such situations happen commonly in distributed computing environments, owing not only to a large variety of possibly distributed tasks but also to other factors such as availability of computing resources and stability of communication channels, etc.

A. The Proposed Parallel Program Architecture

We propose a new architecture of parallel programs, where problem solving is carried out through rule application and rule generation. The architecture is based on the master-worker (or parent-child) model of parallel computing, where one process, referred to as a master (or a parent), solves a problem by distributing some tasks to other processes, referred to as workers (or children). Assume that a specification S is given. The master has a set of equivalent transformation rules (ET rules) with respect to S and its computation state is represented as a set of definite clauses. The master makes computation by successively applying ET rules to clauses in its state. A worker has a rule generator and makes computation by generating ET rules with respect to S. Having an initial set of ET rules, the master (i) selects an ET rule from its rule set and applies the rule, or (ii) sends an atom set to a worker as a request for rule generation, or (iii) receives an ET rule from a worker and adds it to the rule set. Based on the given information from the master (a selected atom set), a worker generates an ET rule using its rule generator, and returns the rule to the master.

B. Effectiveness of the Proposed Architecture

Using the proposed architecture, the correctness of computation can be guaranteed by combination of correct rules and correct rule generators. From atom sets observed at run-time, a master requests its workers to generate specialized rules on demand. Using specialized rules that are tailored to run-time content of computation states, substantial efficiency improvement of computation in the master process can be achieved, i.e., transformation steps can be reduced and computational explosion can be suppressed. It is difficult to predict which specialized rules should be generated beforehand, and generating all specialized rules in advance at compile time is impractical because there are usually far too many possible states and there are usually many possible specialized rules for each state, only a small part of which is really used. Distributing run-time rule generation to workers releases the master from the task of generating specialized rules, which can take much time even when specific patterns of target atoms are already determined.

The initial rule set of a master is prepared in such a way that it is sufficient for solving a problem, albeit not efficiently, without additional rules obtained from workers. As a consequence, delayed response of workers and communication failure do not affect the completion of a problem solving process. Rules obtained form workers contribute to computation speedup, rather than completion of computation. Since rules have a precise procedural semantics [1], the use of rules as messages returned from workers makes clear the meanings of the messages. A returned rule can be used any time no matter how the state of the master is changed during the course of master-worker interaction. No adjustment of returned rules is required even when the order of rule-generation requests and the order of the arrival of their corresponding returned rules do not coincide.

C. Paper Organization and Notation

The paper progresses from here as follows: Section II establishes a class of specifications considered herein. Section III describes parallel programs in our proposed framework. Section IV presents a scheme for construction of parallel programs based on rule-set generators and rule-generator generators, along with its correctness theorem. Section V provides methods for constructing rule-set generators and rule-generator generators. Section VI illustrates application of our framework to solving a constraint satisfaction problem. Section VII concludes the paper.

The following notation holds thereafter. For any set A, $\mathit{pow}(A)$ denotes the power set of A. For any sets A and B, $\mathit{PartialMap}(A, B)$ denotes the set of all partial mappings from A to B.

II. Specifications

A specification provides background knowledge in a problem domain and defines a set of queries of interest. A specification considered in this paper is formulated using the concepts recalled below.

Assume that an alphabet Δ for first-order logic is given. Let $\mathcal{A}$ and $\mathcal{G}$ be the set of all first-order atomic formulas (atoms) and that of all ground atoms, respectively, on Δ. Let $\mathcal{S}$ denote the set of all substitutions on Δ. A *definite clause* C on Δ is an expression of the form $a \leftarrow \mathit{Bs}$, where $a \in \mathcal{A}$ and Bs is a (possibly empty) finite subset of $\mathcal{A}$. The set $\{a\} \cup \mathit{Bs}$ is denoted by $\mathit{atoms}(C)$; a is called the *head* of C, denoted by $\mathit{head}(C)$; Bs is called the *body* of C, denoted by $\mathit{body}(C)$; and each element of $\mathit{body}(C)$ is called a *body atom* of C. When $\mathit{body}(C) = \varnothing$, C is called a *unit clause*. The set notation is used in the right-hand side of C so as to stress that the order of atoms in $\mathit{body}(C)$ is not important. However, for the sake of simplicity, set braces enclosing body atoms are often omitted; e.g., a definite clause $a \leftarrow \{b_1, \ldots, b_n\}$ is often written as $a \leftarrow b_1, \ldots, b_n$.

A *declarative description* on Δ is a set of definite clauses.[1] The meaning of a declarative description is defined as follows: Given a definite clause C and a set $G \subseteq \mathcal{G}$, let

$$T(C, G) = \{\mathit{head}(C\theta) \mid (\theta \in \mathcal{S}) \ \& \ (\mathit{atoms}(C\theta) \subseteq \mathcal{G}) \ \& \ (\mathit{body}(C\theta) \subseteq G)\}.$$

Let D be a declarative description. A mapping T_D on $\mathit{pow}(\mathcal{G})$ is defined by $T_D(G) = \bigcup_{C \in D} T(C, G)$ for any $G \subseteq \mathcal{G}$. The meaning of D, denoted by $\mathcal{M}(D)$, is then defined as the set $\bigcup_{n=1}^{\infty} T_D^n(\varnothing)$, where $T_D^1(\varnothing) = T_D(\varnothing)$ and $T_D^n(\varnothing) = T_D(T_D^{n-1}(\varnothing))$ for each $n > 1$.

A *specification* on Δ is a pair $S = \langle D, Q \rangle$, where D is a declarative description, representing background knowledge, and Q is set of atoms, representing queries. For any $q \in Q$, the answer to q with respect to S is defined as the set

$$\mathcal{M}(D) \cap \mathit{rep}(q),$$

where for any atom a, $\mathit{rep}(a)$ is the set of all ground instances of a. Let Spec be the set of all such specifications.

III. Parallel Programs

A. Rules and Rule Generators

A *transformation rule* (for short, *rule*) is a relation on definite-clause sets. A rule r transforms a set Cs of definite clauses into another set Cs' of definite clauses if $\langle \mathit{Cs}, \mathit{Cs}' \rangle \in r$. Let Rule be the set of all such rules.

Assume that a set R of rules is given. To compute the answer to a query q using R, a singleton definite-clause set

$$\mathit{Cs}_0 = \{\phi(q) \leftarrow q\}$$

is constructed, where ϕ is a bijective mapping that associates with each atom $a \in \mathcal{A}$ an atom obtained from a by replacing its predicate symbol with a new predicate symbol. Then a transformation sequence

$$[\mathit{Cs}_0, \mathit{Cs}_1, \ldots, \mathit{Cs}_m]$$

is constructed such that (i) for each i, $\langle \mathit{Cs}_i, \mathit{Cs}_{i+1} \rangle \in r$ for some rule $r \in R$, and (ii) Cs_m is a set of unit clauses.

A *rule generator* is a partial mapping from $\mathit{pow}(\mathcal{A})$ to Rule. When an atom set $A \subseteq \mathcal{A}$ is given as input, a rule generator yields a rule for transforming some definite clauses whose bodies contain instances of A.

B. Parallel Programs

Assume that

- a master has n workers $w_1, \ldots, w_n$,
- R_0 is a set of rules, and
- for each $i \in \{1, \ldots, n\}$, gen_i is a rule generator.

The pair $\langle R_0, [\mathit{gen}_1, \ldots, \mathit{gen}_n] \rangle$ determines a parallel procedure, consisting of $n + 1$ processes, which is described below. Assume that an input query $q \in \mathcal{A}$ is given.

- *The Master Process:* The master has a state $\langle \mathit{Cs}, R \rangle$, where Cs is a set of definite clauses and R is a set of rules, and it works as follows:
  - Initially, the master sets
    - $\mathit{Cs} = \{(\phi(q) \leftarrow q)\}$, and
    - $R = R_0$.
  - If Cs contains some non-unit clause, then the master performs one of the following operations nondeterministically whenever possible:
    - Select a non-unit clause C from Cs, select a rule from R, and update Cs by applying the selected rule to C.

[1] We call a set of definite clauses a *declarative description* in order to emphasize that no procedural meaning of it is considered.

  - $*$ Select an atom set A from the body of one clause in *Cs* and send it to a worker.
  - $*$ Receive a rule r from a worker and add r to R.
- – If *Cs* contains only unit clauses, then the master outputs the set

$$\bigcup\{rep(\phi^{-1}(a)) \mid (a \leftarrow) \in Cs\}$$

as the computed answer.

- *A Worker Process:* A worker w_i has a rule generator gen_i and has a buffer storing atom sets received from the master. At any time, it performs the following operations sequentially:
  1) Select one atom set A from the buffer.
  2) Generate a rule $r = gen_i(A)$.
  3) Return r to the master.

With a design of more detailed control mechanism, the procedure thus obtained is converted into a program in a lower-level language, making optimization of state representation and memory usage. The lower-level implementation part is however outside the scope of this paper, and the pair $\langle R_0, [gen_1, \ldots, gen_n]\rangle$ is also regarded as a *parallel program.*

IV. A Parallel-Program Construction Scheme

A scheme for constructing parallel programs using rule-set generators and rule-generator generators is next described.

A. Rule-Set Generators and Rule-Generator Generators

First, a rule-set generator and a rule-generator generator are introduced:

- A *rule-set generator* generates a set of rules from a given specification. It is formalized as a mapping from Spec to *pow*(Rule).
- A *rule-generator generator* generates from an input specification a rule generator (which is a partial mapping from $pow(\mathcal{A})$ to Rule). It is formalized as a mapping from Spec to $PartialMap(pow(\mathcal{A}), \text{Rule})$.

Based on the concept of an equivalent transformation rule, the correctness of a rule-set generator and that of a rule-generator generator are defined:

- A rule r is an *equivalent transformation rule* (*ET rule*) with respect to a declarative description D iff for any $\langle Cs, Cs'\rangle \in r$, $\mathcal{M}(D \cup Cs) = \mathcal{M}(D \cup Cs')$.
- A rule-set generator *RSG* is *correct* iff for any specification $S = \langle D, Q\rangle \in$ Spec, every rule in $RSG(S)$ is an ET rule with respect to D.
- A rule-generator generator *RGG* is *correct* iff for any specification $S = \langle D, Q\rangle \in$ Spec and any $A \subseteq \mathcal{A}$, if $RGG(S)(A)$ is defined, then it is an ET rule with respect to D.

B. A General Parallel-Program Construction Scheme

Construction of a correct rule-set generator and that of a correct rule-generator generator provide a general groundwork for constructing correct parallel programs from specifications, using the following parallel-program construction scheme.

1) Construct a correct rule-set generator *RSG*.
2) Construct correct rule-generator generators RGG_1, $\ldots, RGG_n$.
3) From a given a specification $S \in$ Spec, construct a parallel program $\langle R_0, [gen_1, \ldots, gen_n]\rangle$ as follows:
   - $R_0 = RSG(S)$.
   - For each $i \in \{1, \ldots, n\}$, $gen_i = RGG_i(S)$.

Even when only one rule-generator generator, say *RGG*, is constructed at Step 2, i.e., $RGG_1 = \cdots = RGG_n = RGG$, n workers can still be useful for parallel processing since there are many possible different atom sets to be distributed to workers. Given a specification S and different atom sets A_1, $\ldots, A_n$, workers with the same rule generator $RGG(S)$ may produce different ET rules $RGG(S)(A_1), \ldots, RGG(S)(A_n)$.

As a larger number of mutually independent rule-generator generators are available, a larger number of effective workers can be used, potentially yielding more efficient parallel computation. The number of effective workers is evaluated by the multiplication of (i) the number of mutually independent rule-generator generators and (ii) the number of atom sets to be sent to workers.

It is shown in [4] that the correctness of a rule-set generator and that of a rule-generator generator together provide a sufficient condition for a guarantee of the correctness of a resulting parallel program. More precisely:

Theorem 1: Suppose that *RSG* is a correct rule-set generator and $RGG_1, \ldots, RGG_n$ are correct rule-generator generators. Then for any $S \in$ Spec, the parallel program

$$\langle RSG(S), [RGG_1(S), \ldots, RGG_n(S)]\rangle$$

is correct with respect to S.

V. Constructing Rule-Set Generators and Rule-Generator Generators

There are many possible correct rule-set generators and many possible correct rule-generator generators. A large variety of correct parallel programs can thus be constructed using the scheme of Section IV. At present, several methods exist for constructing correct rule-set generators and correct rule-generator generators, some of which are described in this section.

A. Meta-Computation-Based Generators

Meta-computation [2], [3] is a general purpose method for generating ET rules from a specification. The method takes a declarative description D and an atom set $A \subseteq \mathcal{A}$ as input, and produces a nonempty output set of ET rules with respect to D for transforming some definite clauses whose bodies contain instances of A. When A is a singleton set containing the most general atom with respect to some predicate p,[2] the output rule set includes the general unfolding rule for p with respect to D.

Using meta-computation, a rule-set generator *RSG* is constructed as follows: For any $S = \langle D, Q \rangle \in$ SPEC,

- generate a set $\{A_1, \ldots, A_q\}$ of atom sets from Q,
- for each $i \in \{1, \ldots, q\}$, construct a rule set R_i from D and A_i by meta-computation, and
- produce $RSG(S) = R_1 \cup \cdots \cup R_q$.

Similarly, using meta-computation, a rule-generator generator *RGG* is constructed as follows: For any $S = \langle D, Q \rangle \in$ SPEC and any atom set $A \subseteq \mathcal{A}$,

- construct a rule set R from D and A by meta-computation,
- select a rule r from R, and
- produce $RGG(S)(A) = r$.

B. Rule Generation Based on Common Specializers

Let $S = \langle D, Q \rangle \in$ SPEC and a singleton atom set $\{a\} \subseteq \mathcal{A}$ be given. Assume that $v_1, \ldots, v_q$ are all the variables that occur in a. A rule can be generated from S and $\{a\}$ based on common specializers as follows:

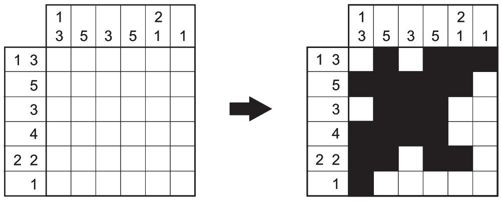
- Calculate the set $G = \{a\theta \mid (\theta \in \mathcal{S}) \ \&\ (a\theta \in \mathcal{M}(D))\}$.
- For each $i \in \{1, \ldots, q\}$, find a common ground term for v_i with respect to G, i.e., find a ground term t_i such that for any $\rho \in \mathcal{S}$, if $a\rho \in G$, then $v_i\rho = t_i$.[3]
- Let E be the sequence of all equality atoms $=(v_i, t_i)$ such that t_i is a common ground term for v_i with respect to G.
- Assuming that the sequence E thus obtained is $[e_1, \ldots, e_{q'}]$, where $q' \leq q$, construct a rule

  $$a \Rightarrow \{e_1, \ldots, e_{q'}\}, a.$$

  This rule is applicable to a definite clause C if there exist $\theta \in \mathcal{S}$ and an atom $b \in body(C)$ such that θ contains only bindings for variables occurring in a and $a\theta = b$. When applied, the rule specializes C by the evaluation of the equality atoms $e_1\theta, \ldots, e_{q'}\theta$.[4]

VI. An Example

To illustrate application of our framework, a Pic-a-Pix puzzle[5] (Oekaki Logic or Paint by Numbers) of a fixed size $m \times n$ is used as an example problem.



Problem

Solution

Figure 1. A Pic-a-Pix puzzle.

C_1: $pat([], *Z) \leftarrow zeros(*Z).$
C_2: $pat([*a|*X], *Y) \leftarrow zeros(*Z), ones(*a, *A), pat(*X, *B), apps([*Z, *A, *B], *Y).$
C_3: $zeros([0]) \leftarrow.$
C_4: $zeros([0|*X]) \leftarrow zeros(*X).$
C_5: $ones(0, []) \leftarrow.$
C_6: $ones(*n, [1|*Y]) \leftarrow >(*n, 0), subtr(*n, 1, *m), ones(*m, *Y).$
C_7: $apps([], []) \leftarrow.$
C_8: $apps([*a|*X], *Y) \leftarrow apps(*X, *Z), app(*a, *Z, *Y).$
C_9: $app([], *Y, *Y) \leftarrow.$
C_{10}: $app([*a|*X], *Y, [*a|*Z]) \leftarrow app(*X, *Y, *Z).$

Figure 2. Representing background knowledge for Pic-a-Pix puzzles.

A. Problem Representation and Formulating a Specification

First, consider the Pic-a-Pix puzzle of size 6×6 in Fig. 1. It consists of a blank grid and clues, i.e., block patterns, on the left of every row and on the top of every column, with the goal of painting blocks in each row and column so that their length and order correspond to the patterns and there is at least one empty square between adjacent blocks.

The puzzle in Fig. 1 is represented by a *Pic-a-Pix*-atom

$$\begin{aligned} Pic\text{-}a\text{-}Pix(&[[1,3],[5],[3],[4],[2,2],[1]],\\ &[[1,3],[5],[3],[5],[2,1],[1]],\\ &*mat), \end{aligned}$$

where $*mat$ is a variable. Other 6×6 Pic-a-Pix puzzles are represented as *Pic-a-Pix*-atoms in a similar way. Let $Q_{6\times6}$ be the set of all such *Pic-a-Pix*-atoms. The specification for this class of puzzles, denoted by $S_{6\times6}$, is defined by

$$S_{6\times6} = \langle \{C_0\} \cup D_{\text{LINK}} \cup D_{\text{PIC}} \cup D_{\text{BLT}}, Q_{6\times6} \rangle,$$

where C_0 is the definite clause

$$\begin{aligned} C_0: \quad Pic\text{-}a\text{-}Pix(*t, *y, *mat) \leftarrow \\ &=(*t, [*t1, *t2, *t3, *t4, *t5, *t6]),\\ &=(*y, [*y1, *y2, *y3, *y4, *y5, *y6]),\\ &matrix(*t, *y, *mat), trans(*mat, *tam),\\ &pairing(*t, *mat), pairing(*y, *tam), \end{aligned}$$

D_{LINK} is a declarative description that provides the definitions of *matrix*, *trans*, and *pairing*, D_{PIC} consists of the definite clauses C_1–C_{10} in Fig. 2, and D_{BLT} provides the definitions of built-in predicates, such as $>$ and *subtr*.

[2] Given an m-ary predicate p, the most general atom with respect to p is $p(v_1, \ldots, v_m)$, where the v_i are mutually different variables.

[3] A common ground term for v_i with respect to G is unique if it exists.

[4] When $q' = 0$, the rule does not make clause specialization and it is not used in a master process.

[5] http://www.conceptispuzzles.com.

$$
\begin{aligned}
C'_0\colon\ & \mathit{Pic\text{-}a\text{-}Pix}([*t1, *t2, *t3, *t4, *t5, *t6],\\
&\quad [*y1, *y2, *y3, *y4, *y5, *y6],\\
&\quad [[*a11, *a12, *a13, *a14, *a15, *a16],\\
&\quad\ [*a21, *a22, *a23, *a24, *a25, *a26],\\
&\quad\ [*a31, *a32, *a33, *a34, *a35, *a36],\\
&\quad\ [*a41, *a42, *a43, *a44, *a45, *a46],\\
&\quad\ [*a51, *a52, *a53, *a54, *a55, *a56],\\
&\quad\ [*a61, *a62, *a63, *a64, *a65, *a66]])\\
& \leftarrow\\
& \mathit{pat}(*t1, [0, *a11, *a12, *a13, *a14, *a15, *a16, 0]),\\
& \mathit{pat}(*t2, [0, *a21, *a22, *a23, *a24, *a25, *a26, 0]),\\
& \mathit{pat}(*t3, [0, *a31, *a32, *a33, *a34, *a35, *a36, 0]),\\
& \mathit{pat}(*t4, [0, *a41, *a42, *a43, *a44, *a45, *a46, 0]),\\
& \mathit{pat}(*t5, [0, *a51, *a52, *a53, *a54, *a55, *a56, 0]),\\
& \mathit{pat}(*t6, [0, *a61, *a62, *a63, *a64, *a65, *a66, 0]),\\
& \mathit{pat}(*y1, [0, *a11, *a21, *a31, *a41, *a51, *a61, 0]),\\
& \mathit{pat}(*y2, [0, *a12, *a22, *a32, *a42, *a52, *a62, 0]),\\
& \mathit{pat}(*y3, [0, *a13, *a23, *a33, *a43, *a53, *a63, 0]),\\
& \mathit{pat}(*y4, [0, *a14, *a24, *a34, *a44, *a54, *a64, 0]),\\
& \mathit{pat}(*y5, [0, *a15, *a25, *a35, *a45, *a55, *a65, 0]),\\
& \mathit{pat}(*y6, [0, *a16, *a26, *a36, *a46, *a56, *a66, 0]).
\end{aligned}
$$

Figure 3. A clause obtained by unfolding C_0.

B. Determining the Forms of Exchange Information

The clause C_0 is unfolded using D_{LINK}, resulting in the clause C'_0 in Fig. 3, which contains $6 + 6$ *pat*-atoms in its body. A copy of each of these *pat*-atoms is then added to the body of C'_0. The predicate *pat:c* is used for denoting a copy.[6] The clause thus obtained is

$$\hat{C}'_0:\quad \mathit{head}(C'_0) \leftarrow \mathit{body}(C'_0) \cup \mathit{copy}(\mathit{body}(C'_0)),$$

where $\mathit{copy}(\mathit{body}(C'_0))$ is the set consisting of the 6+6 added *pat:c*-atoms. These *pat:c*-atoms are designated as messages that a master sends to workers. The original specification $S_{6\times 6}$ is then transformed into the specification

$$S'_{6\times 6} = \langle \{\hat{C}'_0, C'_1, C'_2\} \cup D_{\text{PIC}} \cup D_{\text{BLT}}), Q_{6\times 6}\rangle,$$

where C'_1 and C'_2 are the definite clauses obtained from C_1 and C_2, respectively, by replacing the predicate *pat* with the predicate *pat:c*.

C. Constructing a Rule Set Using Rule-Set Generators

Using a meta-computation-based rule-set generator, all rules in Fig. 4 except r_6 and also all rules in Fig. 5 are generated. The rule r_6 can be generated using another rule-set generator, based on a method of finding a common specialization from definite clauses. An unfolding rule corresponding to $\hat{C}'_0$ is also generated using a meta-computation-based rule-set generator.

The rules r_1–r_{17} are specialized rules; they are applicable to clauses whose bodies contain atoms having certain specific patterns. For example, r_1 (respectively, r_2) is applicable to any clause containing in its body a *pat*-atom the first argument of which is an empty (respectively, nonempty) list. Both r_1 and r_2 have no execution part—they make transformation merely by replacement of body atoms. The rules r_3–r_6 are specialized rules for *zeros*-atoms. Since the evaluation of the atom *false* fails, r_3 always makes clause removal when it is applied. Each of r_5 and r_6 contains both an execution part and a replacement part. By using a *pvar*-atom to constrain its applicability, r_6 is only applicable to a *zeros*-atom whose argument is a variable. The rule r_{18} is a general rule; it is applicable to any clause whose body contains any arbitrary *app*-atom. Since r_{18} has two bodies in its right side, its application typically splits a clause into two clauses.

$$
\begin{aligned}
r_1\colon\ & \mathit{pat}([], *Z) \Rightarrow \mathit{zeros}(*Z).\\
r_2\colon\ & \mathit{pat}([*a|*X], *Y)\\
& \Rightarrow \mathit{zeros}(*Z), \mathit{ones}(*a, *A), \mathit{pat}(*X, *B), \mathit{apps}([*Z, *A, *B], *Y).\\
r_3\colon\ & \mathit{zeros}([]) \Rightarrow \{\mathit{false}\}.\\
r_4\colon\ & \mathit{zeros}([*a]) \Rightarrow \{=(*a, 0)\}.\\
r_5\colon\ & \mathit{zeros}([*a, *b|*X]) \Rightarrow \{=(*a, 0)\}, \mathit{zeros}([*b|*X]).\\
r_6\colon\ & \mathit{zeros}(*X), \{\mathit{pvar}(*X)\} \Rightarrow \{=(*X, [0|*Y])\}, \mathit{zeros}(*X).\\
r_7\colon\ & \mathit{ones}(0, *Y) \Rightarrow \{=(*Y, [])\}.\\
r_8\colon\ & \mathit{ones}(*n, *X), \{>(*n, 0)\}\\
& \Rightarrow \{=(*X, [1|*Y]), \mathit{subtr}(*n, 1, *m)\}, \mathit{ones}(*m, *Y).\\
r_9\colon\ & \mathit{apps}([], *Y) \Rightarrow \{=(*Y, [])\}.\\
r_{10}\colon\ & \mathit{apps}([*a|*X], *Y) \Rightarrow \mathit{apps}(*X, *Z), \mathit{app}(*a, *Z, *Y).\\
r_{11}\colon\ & \mathit{app}(*X, *Y, []) \Rightarrow \{=(*X, []), =(*Y, [])\}.\\
r_{12}\colon\ & \mathit{app}(*X, [], *Z) \Rightarrow \{=(*X, *Z)\}.\\
r_{13}\colon\ & \mathit{app}([], *Y, *Z) \Rightarrow \{=(*Y, *Z)\}.\\
r_{14}\colon\ & \mathit{app}([*a|*X], *Y, *Z) \Rightarrow \{=(*Z, [*a|*Z1])\}, \mathit{app}(*X, *Y, *Z1).\\
r_{15}\colon\ & \mathit{app}(*X, [*a|*Y], [*b|*Z]), \{\mathit{neq}(*a, *b)\}\\
& \Rightarrow \{=(*X, [*b|*X1])\}, \mathit{app}(*X1, [*a|*Y], *Z).\\
r_{16}\colon\ & \mathit{app}(*X, [*a1, *a2|*Y], [*b1, *b2|*Z]), \{\mathit{neq}(*a2, *b2)\}\\
& \Rightarrow \{=(*X, [*b1|*X1])\}, \mathit{app}(*X1, [*a1, *a2|*Y], [*b2|*Z]).\\
r_{17}\colon\ & \mathit{zeros}([*a|*X]), \mathit{app}(*X, [1|*M], [1|*R])\\
& \Rightarrow \{=(*a, 0), =(*X, []), =(*M, *R)\}.\\
r_{18}\colon\ & \mathit{app}(*X, *Y, *Z)\\
& \Rightarrow \{=(*X, []), =(*Y, *Z)\};\\
& \Rightarrow \{=(*X, [*a|*X1]), =(*Z, [*a|*Z1])\}, \mathit{app}(*X1, *Y, *Z1).
\end{aligned}
$$

Figure 4. Rules for solving Pic-a-Pix puzzles.

$$
\begin{aligned}
r_{19}\colon\ & \mathit{pat{:}c}(*R, [0, 0|*S]) \Rightarrow \mathit{pat{:}c}(*R, [0|*S]).\\
r_{20}\colon\ & \mathit{pat{:}c}([], [0, 1|*S]) \Rightarrow \{false\}.\\
r_{21}\colon\ & \mathit{pat{:}c}([], [0]) \Rightarrow.\\
r_{22}\colon\ & \mathit{pat{:}c}([*n|*R], [0, 1, 0|*S]), \{>(*n, 1)\} \Rightarrow \{false\}.\\
r_{23}\colon\ & \mathit{pat{:}c}([*n|*R], [0, 1, 1|*S]), \{>(*n, 1)\}\\
& \Rightarrow \{\mathit{subtr}(*n, 1, *m)\}, \mathit{pat{:}c}([*m|*R], [0, 1|*S]).\\
r_{24}\colon\ & \mathit{pat{:}c}([1|*R], [0, 1, *y|*S]) \Rightarrow \{=(*y, 0)\}, \mathit{pat{:}c}(*R, [0|*S]).
\end{aligned}
$$

Figure 5. ET rules for removing useless parts of *pat:c*-atoms.

D. Constructing a Rule Generator

Using rule generation based on common specializers described in Section V-B, a rule generator employed by a worker is constructed. To illustrate, suppose that a singleton

[6] For example, the copy of the first body atom of C'_0 is $\mathit{pat{:}c}(*t1, [0, *a11, *a12, *a13, *a14, *a15, *a16, 0])$.

atom set $\{a\}$, where

$$a \;=\; \mathit{pat{:}c}([1,3],[0,*x1,*x2,*x3,*x4,*x5,*x6,0]),$$

is given to a worker. The worker applies the rules in Fig. 4 to make a transformation sequence producing the set G consisting of all ground instances of a that belong to $\mathcal{M}(\{\hat{C}'_0, C'_1, C'_2\} \cup D_{\mathrm{PIC}} \cup D_{\mathrm{BLT}})$. The resulting set G consists of the following three ground atoms:

- $\mathit{pat{:}c}([1,3],[0,1,0,1,1,1,0,0])$
- $\mathit{pat{:}c}([1,3],[0,1,0,0,1,1,1,0])$
- $\mathit{pat{:}c}([1,3],[0,0,1,0,1,1,1,0])$

From the common part of these atoms, the sequence of equality atoms $[=(*x4,1), =(*x5,1)]$ is obtained. Accordingly, the rule

$$\begin{aligned}&\mathit{pat{:}c}([1,3],[0,*x1,*x2,*x3,*x4,*x5,*x6,0])\\ &\Rightarrow \{=(*x4,1), =(*x5,1)\},\\ &\quad \mathit{pat{:}c}([1,3],[0,*x1,*x2,*x3,*x4,*x5,*x6,0])\end{aligned}$$

is generated.

E. Parallel Computation

When the master receives an input problem $q \in Q_{6\times 6}$, it creates an initial clause set $Cs_0 = \{\phi(q) \leftarrow q\}$. When the computation starts, the master transforms $\phi(q) \leftarrow q$ using the unfolding rule corresponding to $\hat{C}'_0$. This transformation yields a definite clause with only *pat*-atoms and *pat:c*-atoms in its body. Using the rules in Fig. 4, which are generated from D_{PIC}, *pat*-atoms are successively transformed. By application of the rules r_1 and r_2, *pat*-atoms are all replaced with some other atoms, while *pat:c*-atoms do not disappear. The single-body rules r_1–r_{17} are given priority over the multi-body rule r_{18}. When no single-body rule is applicable, the master uses r_{18}, which increases the number of clauses.

Supposing that the master only applies the rules in its initial rule set without requesting any worker to generate any additional specialized rule, the answer to the puzzle in Fig. 1 is obtained after 10,789 rule application steps, 1,520 of which are clause-splitting steps. In comparison, when workers are used to generate specialized rules on demand based on the proposed architecture, the total number of rule application steps in the master process reduces from 10,789 to 512 in our experiment, with the number of clause-splitting steps reducing from 1,520 to zero.

Employment of specialized rules obtained by rule generation on demand based on run-time content of computation states usually decreases problem-solving time greatly. Compared to application of an existing rule, generation of a new rule itself may take much time. By distributing the tasks of run-time rule generation to workers, the master does not bear the cost of rule generation and, therefore, the overall computation time in the master process substantially reduces.

VII. Conclusions

We establish a theory of direct connection between specifications and correct parallel programs, which shows a sharp contrast to the usual parallel logic programming approaches ([5], [6], [7]), where a human programmer constructs a parallel program based on a specification without a guarantee of correctness. A parallel program in our framework consists of ET rules and rule generators. If all ET rules and all rule generators in a parallel program are correct, then the program is correct. Since there are a large variety of ET rules and rule generators, our framework provides a large space of correct parallel programs, which widens the possibility of finding an efficient program with a relatively small cost. ET rules and rule generators are constructed not only by human programmers, but also by automatic rule generators. A program construction scheme based on rule-set generators and rule-generator generators is described. As long as correct rule-set generators and rule-generator generators are used, a resulting program always yields correct computation and, consequently, verification of the obtained program, which is usually a very expensive task, is not necessary.

References

[1] K. Akama and E. Nantajeewarawat, Formalization of the Equivalent Transformation Computation Model, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 10: 245–259, 2006.

[2] K. Akama, E. Nantajeewarawat, and H. Koike, Program Synthesis Based on the Equivalent Transformation Computation Model, Proc. the 12th International Workshop on Logic Based Program Synthesis and Transformation, Madrid, Spain, pp. 285–304, 2002.

[3] K. Akama, E. Nantajeewarawat, and H. Koike, Program Generation in the Equivalent Transformation Computation Model Using the Squeeze Method, *Perspectives of System Informatics*, Lecture Notes in Computer Science, Vol. 4378, pp. 41–54, Springer-Verlag, Berlin Heidelberg, 2007.

[4] K. Akama, E. Nantajeewarawat, and H. Koike, Constructing Parallel Programs Based on Rule Generators, Technical Report, Hokkaido University, Sapporo, Japan, 2011.

[5] J. Chassin de Kergommeaux and P. Codognet, Parallel Logic Programming Systems, *ACM Computing Surveys*, 26: 295–336, 1994.

[6] G. Gupta, E. Pontelli, K. Ali, M. Carlsson, and M. Hermenegildo, Parallel Execution of Prolog Programs: a Survey, *ACM Transactions on Programming Languages and Systems*, 23: 472–602, 2001.

[7] V. Santos Costa, Parallelism and Implementation Technology for Logic Programming Languages, in *Encyclopedia of Computer Science and Technology*, Vol. 42, pp. 197–237, Marcel Dekker Inc., New York, 2000.

Impact of User Concurrency in Commonly Used Open Geospatial Consortium Map Server Implementations

Joan Masó, Paula Díaz, Xavier Pons, José L. Monteagudo-Pereira, Joan Serra-Sagristà, Francesc Aulí-Llinàs
Universitat Autònoma de Barcelona, Spain
joan.maso@uab.cat, paula.diaz@uab.cat, xavier.pons@uab.cat, jlino@deic.uab.cat, joan.serra@uab.cat, fauli@deic.uab.cat

***Abstract*—In emergency situations, it is of paramount importance that accurate assessments showing what is happening in the field, and when and where it is happening, are distributed quickly. This increases the aid workers' awareness of the situation and can be used to organize the workers more efficiently. The increasing number of satellite images available means that new data can be obtained rapidly and the information can be kept constantly up to date. These data can be distributed easily using open standards over the Internet. In a large post-disaster event, the demand for information increases dramatically, which can negatively impact the performance of the services provided. Here, we assess seven of the most popular server solutions (GeoServer, MapServer, MiraMon Map Server, Express Server, ArcGIS Server, TileCache and GeoWebCache) for map service standards (WMS, WMTS, WMTS-C, TMS), and compare their response times, user functionalities and usability.**

***Keywords-server; WMS; tile; response time; cluster; performance*.**

I. INTRODUCTION

Nowadays, there is an increasingly large amount of data, software and geographic standards available (public, private and voluntary), which allow satellite data to be used in a wider range of consolidated and specialized areas and applications. Current space technologies, such as meteorological and earth observation satellites integrated in global networks like GMES, communication satellites and Global Navigation Satellite Systems (GNSS) combined with Geographical Information Systems (GIS) [1], hazard modeling and analysis have also contributed to this increase in applications and data. Nevertheless, better spatial, temporal (synergies, constellations, etc.) and spectral resolutions of remote sensing imagery generate a huge amount of data that is difficult to store, discover, analyze and distribute. Heaps of tapes, CDs and hard drives full of data have been replaced by web-based data dissemination infrastructures that make searching and discovery easier. Web portals and clearinghouses increasingly implement standardized protocols and are integrated into a larger System of Systems, like GEOSS [19].

Despite the number of map server implementations that claim to be the fastest and the most robust on the market, there are few studies that apply rigorous metrics to determine the real performance of the servers or compare strategies to increase their performance.

This paper is an extension of a previous article [11] that evaluates the efficiency and possibilities of several map servers (i.e., MapServer [7], GeoServer [4], MiraMon Map Server [18], Express Server [6], ArcGIS Server [3], TileCache [13, 8] and GeoWebCache [15]) that implement international standards (e.g., Web Map Server (WMS [2]), Web Map Server – Cache (WMS-C [13]), Web Map Tile Server (WMTS [9])) connected to satellite image repositories. This research was carried out in the context of GEO-PICTURES, an acronym for *GMES and Earth Observation combined with Position based Image and sensor Communications, Technology for Universal Rescue, Emergency and Surveillance*. GEO-PICTURES is an EU FP7 SPACE project that aims to integrate satellite imagery into in-situ sensors and geo-tagged media (photos and video) to create a tool for decision making in emergency disaster situations. The complete GEO-PICTURES solution covers the capture, transmission, and analysis of data, which is re-elaborated and re-distributed to the aid forces as well as to the general public using several web platforms.

This article begins with the description of the materials and methodology used to perform this study, followed by a through explanation of the tests applied to the performance of the servers. Here an evaluation of concurrent requests to a single server and to a cluster of servers is done. The article continues with the comparison of different standards, this is what it is called: tiling the request and response. Finally, it concludes with a section where the most relevant results are discussed.

II. MATERIALS AND METHODOLOGY

Trials were performed with 22 GeoEye-1 (Orthorectified GeoTIFF; provided by Google [5]) imagery datasets that form a 4Gb raster of 40994 x 57392 pixels, covering Port-au-Prince and surroundings, on 16 January, 2010, 3 days after the earthquake. The influence of scale, intensive client use, and image size and format (JPG, GIF and PNG) were studied for the WMS, WMS-C and TMS [16, 17] protocols. As possible solutions to concurrent requests, we evaluated the efficiency of the Internet cache as well as a cluster of servers in an NLB (Network Load Balance) configuration. The seven servers were set with the minimum configuration required to be run, i.e., without any extra preparation of the data. In order to shield probabilistic error caused by network latency and other uncertain factors, the products were considered to be deployed and requested by local clients [14].

In order to guarantee the comparability of the results, the seven server software (Table 1) were installed on the same computer, which acted as a common server (Intel® Core™ i3 CPU 540 @ 3.07 GHz, 3.06 GHz, 2.92 GB de RAM. Microsoft Windows XP Professional, version 2002 Service Pack 3). Requests were randomly generated from clients in the local network.

TABLE I. LIST OF THE SERVERS EVALUATED AND THEIR SPECIFICATIONS.

Server	Specifications	Date
MapServer	version 5.6.3 over Apache web server version 2.2.15	April 20, 2010
GeoServer	version 1.7.2 over Jetty web server version 6.1.8	January 19, 2008
ArcGIS Server	version 9.3.1 over Internet Information Service version 5.1	January 1, 2009
Express Server	version 6.1 over Internet Information Service version 5.1	July 1, 2008
MiraMon Map Server	version v. 7.0e over Internet Information Service version 5.1	July 26, 2010
TileCache	version 2.11 over Internet Information Service version 5.1	December 22, 2007
GeoWebCache	version 2.0.1 over Jetty web, 'build over GeoServer	January 20, 2010

MapServer, GeoServer and ArcGIS Server can work over the original GeoTIFF images without having to create an image mosaic. MapServer and GeoServer require a shape file to be created with rectangles representing bounding boxes of each raster file (index file), and ArcGIS Server requires Image Definition (ISDef), which makes it possible to use the original GeoTIFF image format provided by GeoEye-1. Express Server requires the image index to be in JPEG2000 or in the MsSID compressed format. MiraMon requires a full automated pre-rendering of the images in several resolutions to set up the images. This takes several minutes to complete and needs up to 33% extra disk space. TileCache and GeoWebCache can create pre-rendered tiles and save them in a cache for further use or generate them on the fly. Automatic on the fly generation is an advantage that can save time when a new layer is set up in an emergency situation.

All studied protocols request maps by creating an URL using specific standardized syntax. This URL is requested from the server and an image is obtained in screen resolution (in the case of error the server sends an exception message). The URL requests were randomly generated by a program and requested using a command line tool (an application called *Wininet*) that waits for the response, saves it and reports the total time spent on this particular communication. The response time was redirected to an archive that was converted into a table of values that were used to create the performance graphics that are shown here. This paper describes the methodology employed and the numerical and graphical performance metrics, and evaluates strategies for improving performance.

The randomly generated URL methodology employed guarantees that speed measures are comparable and independent from the selected bounding box or request sequence. Nevertheless, users in front of a computer screen browsing the maps do not generate random requests but rather they request regions next to the previous ones at the same zoom level (pan) or they zoom in and out in the same region. However, human browsing patterns are out of the scope of this work and will be considered in the future.

III. EVALUATION OF CONCURRENT REQUESTS TO A SINGLE SERVER

One of the main factors that affect the performance of web servers is the concurrency of requests. We measured both the influence of the pixel size and the image size on the response time for WMS requests. More than one hundred different requests were made from up to 6 simultaneous clients. The graph (Figure 1) shows the response time for different pixel size requests.

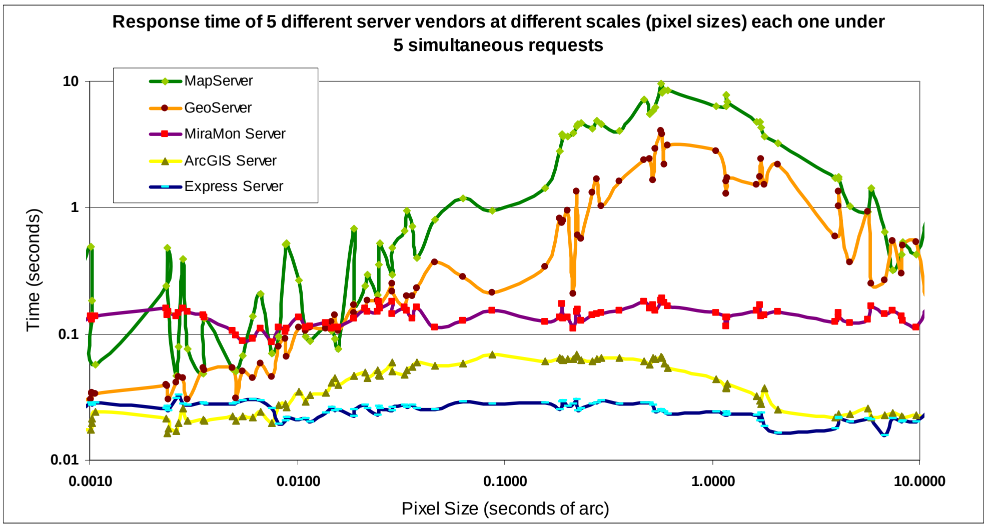


Figure 1. The response times of MapServer, GeoServer, MiraMon Map Server, ArcGIS Server and Express Server in relation to pixel size in concurrent requests from 5 simultaneous clients respectively.

One common aspect of the servers analyzed is that the response time increases as the number of simultaneous requests made by clients to a single server increases. The fastest server is Express Server, probably due to the nature of the wavelet compressed format (MrSID or JPEG2000) that is internally organized in a pyramid of zoom levels of the images used as a database. ArcGIS Server obtains the best results using original GeoTIFF images. MiraMon Map Server obtains intermediate results as it requires a pre-rendering process to generate tiles. MapServer is programmed in C language and GeoServer uses java code. MapServer performs faster when a single client is used, but GeoServer is faster than MapServer for concurrent requests. This could be because java provides better and easier multithread support. There are also small differences in response times depending on the output format requested. JPEG is the fastest format in MapServer, Express Server, ArcGIS Server and MiraMon Map Sever, but PNG is fastest in GeoServer.

A request with a pixel size that generates a map covering a region equivalent to the boundary of the GeoTIFF set (nearly 0.893 seconds of arc in range, width 443) obtains the slowest response time. A map with a smaller pixel size only shows a part of the GeoTIFF set area and obtains a faster response. A map with a larger pixel size leaves some blank areas, and also results in a faster response time.

IV. EVALUATION OF A CLUSTER OF SERVERS

In the current state of maturity of the hardware it is not possible to dramatically increase performance by getting a faster machine, even if you are willing to pay more. Current computer technology has reached a speed limit in CPU processing time and disk speed access. To overcome the performance degradation observed in concurrent requests a possible solution is to set up a cluster of servers that can act as a virtual single server that deals with requests in parallel. We carried out tests comparing a single WMS server (Figure 2) and a WMS computer cluster server (Figure 3), in which 6 computers are able to respond to different clients at the same time as if they were a faster single server. These tests (consisting in the same requests) were carried out with up to 17 simultaneous clients to evaluate the response time of the MiraMon Map Server.

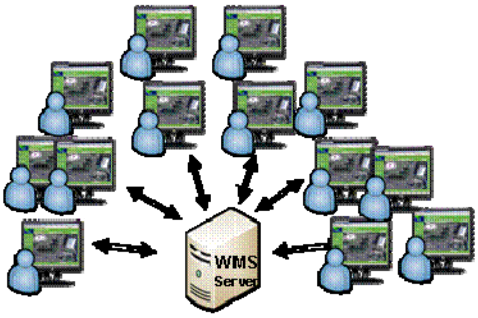


Figure 2. Computer single server structure.

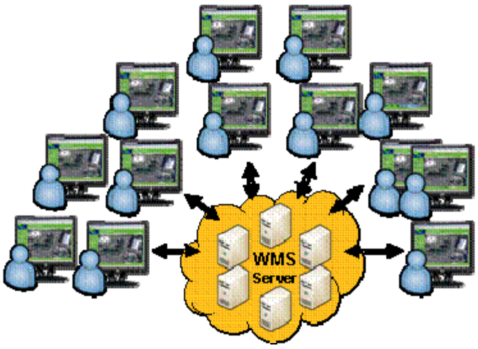


Figure 3. Computer cluster server structure.

ISBN: 978-1-61208-161-8

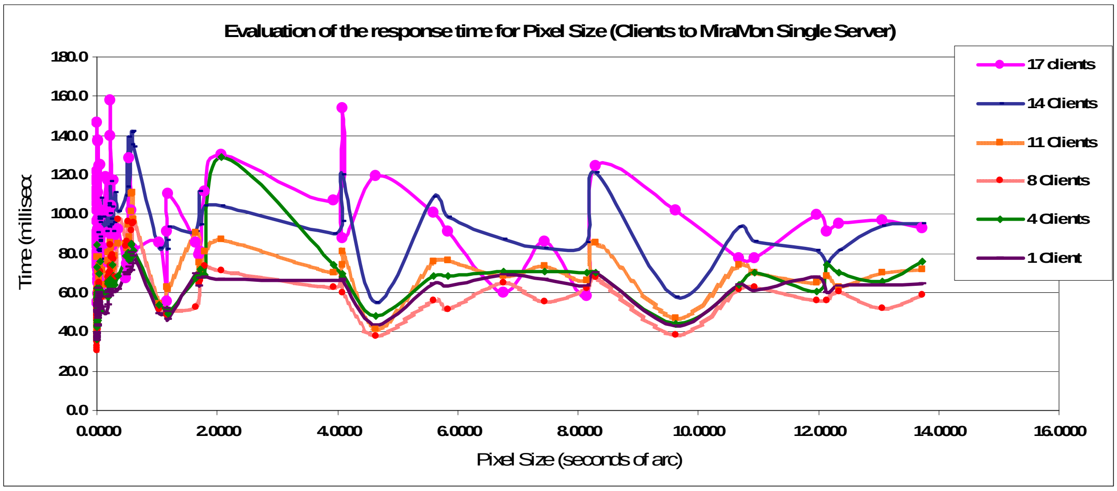


Figure 4. Response time for different concurrent requests for up to 17 clients of a single MiraMon Map Server.

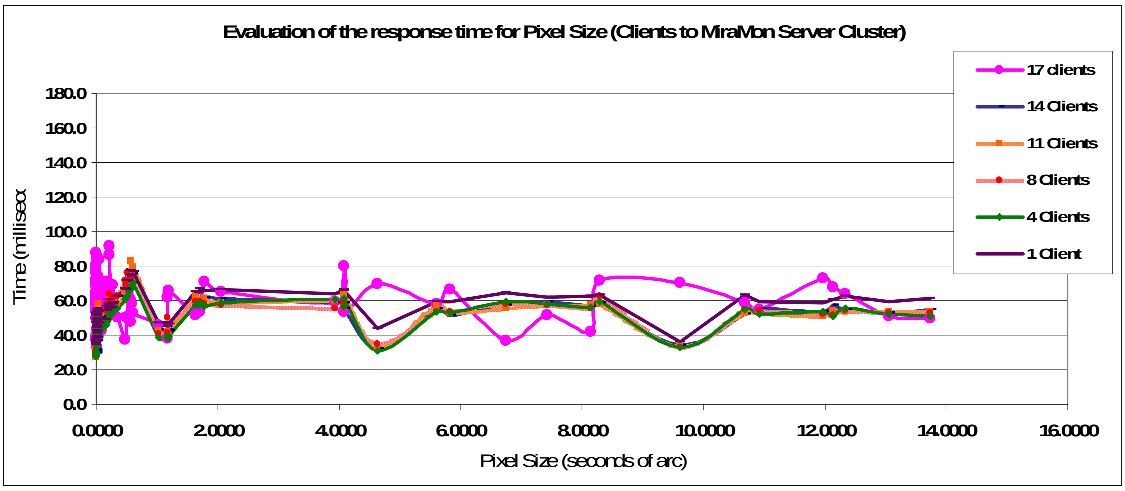


Figure 5. Response time for different concurrent requests for up to 17 clients of a cluster MiraMon Map Server.

Response time measurements comparing a single server (Figure 4) and a six-computer cluster (Figure 5) stressed with multiple client requests show that the response time of the NLB server is much more stable and almost equivalent to the single client stress case, even for 17 simultaneous requests. We expect some degradation if we increase the number of requests further, but fortunately the performance of the NLB cluster can be improved again by aggregating more servers to the cluster (up to 64 in Windows 2003). If we suppose that the performance is linear, this means that this configuration can be scaled to serve at least ~200 simultaneous requests without performance degradation. Note that the response time for these requests is always lower that 0.1 second, so this configuration is equivalent to 2000 requests per second.

V. TILING THE REQUEST AND RESPONSE

In the previous sections, we assumed a common WMS interaction in which a WMS client requests the entire image needed to cover the client viewport in a single piece. Some WMS clients (like OpenLayers) are now able to tile the space in a regular matrix of small pieces [12]. Therefore, several tiles are needed to cover the whole viewport but the client can recycle some tiles when the user moves the view laterally and can also take advantage of the cache mechanisms. However, this strategy can have its drawbacks if the caching mechanism cannot help and the server has not been prepared to manage this situation because, as we have discussed previously, the response time can increase even if each tile is smaller than the whole view. However, users do not perceive this because some tiles get to the client sooner and are shown immediately. This paper clarifies the effects of this approach on a classical WMS server and quantifies the difference between fast full image delivery (WMS) and tiled image delivery (WMS-C and TMS). We also studied improving these situations by applying tile strategies directly to the server, like OSGeo WMS-C and OGC WMTS (10).

We carried out speed metrics in 3 different services for 7 servers: Express Server, ArcGIS Server, MiraMon Map Server, GeoServer, MapServer, TileCache combined with MapServer and GeoWebCache combined with GeoServer. We simulated tiled clients (tiles of 256x256 pixels) that make requests to common WMS (which have no particular strategy for dealing with tiles) in the three following configurations:

- Tiled WMS in unlimited concurrent requests. This consists in requesting all the tiles needed to cover the viewport at the same time. In our case, from 6 to 12 tiles were needed to cover the entire viewport requested. In some cases, this resulted in momentary server saturation (Figure 6), like in MapServer and GeoServer. The three servers with the best performance were Express Server, ArcGIS Server and GeoWebCache.

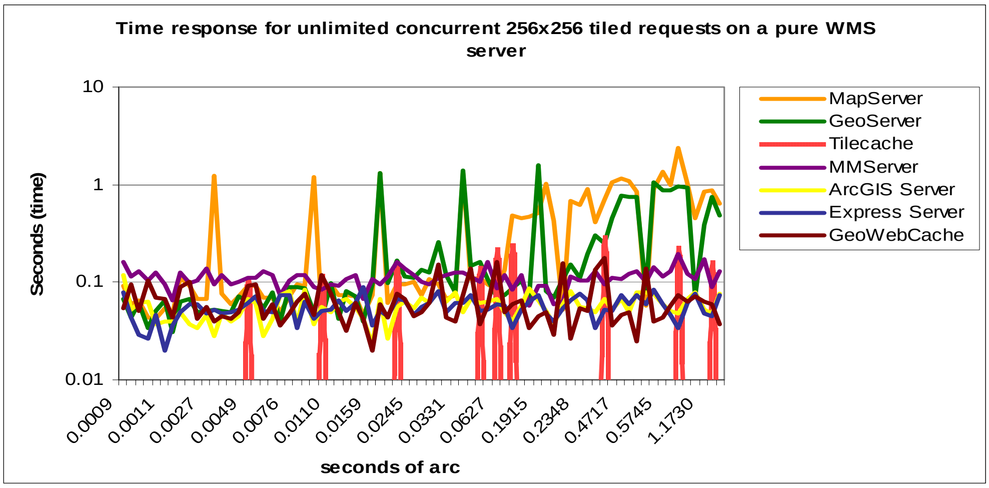


Figure 6 . WMS-C for unlimited concurrent tile requests.

- Tiled WMS in semi-concurrent requests. This consists in limiting simultaneous requests to the maximum number of requests to a server that a web browser allows (e.g., Firefox 3.6 allows 6 simultaneous petitions but Internet Explorer 6.0 only allows 2). In our case, we used a mean value of four tiles at the same time, then we waited until the server finished to request the next four (Figure 7). Some servers performed better compared to the previous case, such as MapServer, GeoServer and MiraMon, while others performed worse. Tiled servers performed better in general.

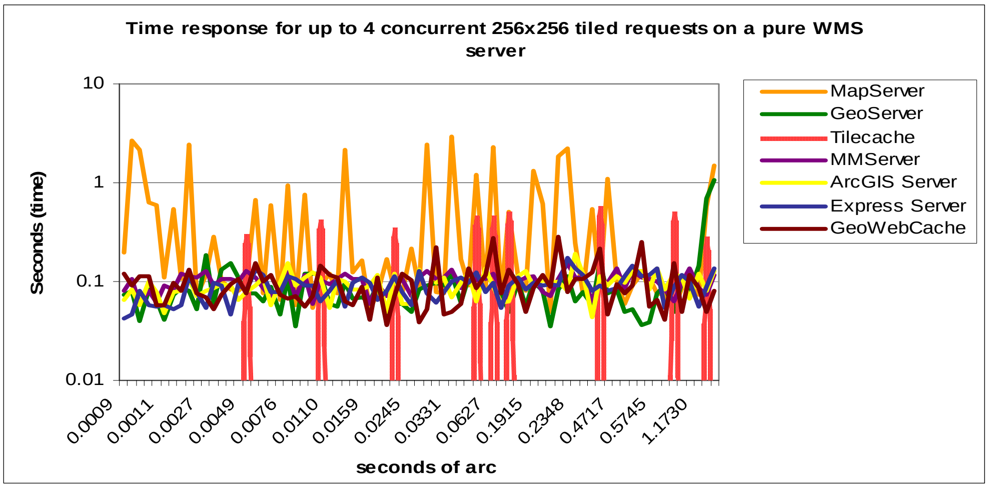


Figure 7.WMS-C for semi-concurrent requests, up to 4 concurrent tile requests.

- Tiled WMS in sequential requests. This consists in requesting each tile after the previous request has been completed (Figure 8). This results in a more stable response time but it is not the optimum situation, especially for GeoServer, MiraMon Server and in some cases ArcGIS Server. GeoWebCache has the best performance.

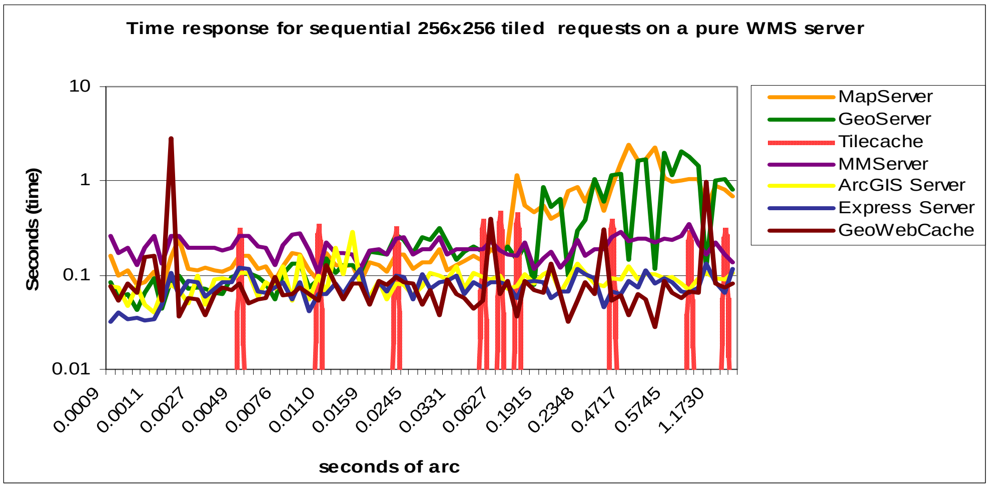


Figure 8. WMS-Cached for sequential requests.

Finally, we compared these three configurations with a regular WMS full viewport image (Figure 9), and evaluated performance degradation. TileCache and GeoWebCache are not represented in Figure 9 because these servers are not able to respond to a full WMS viewport.

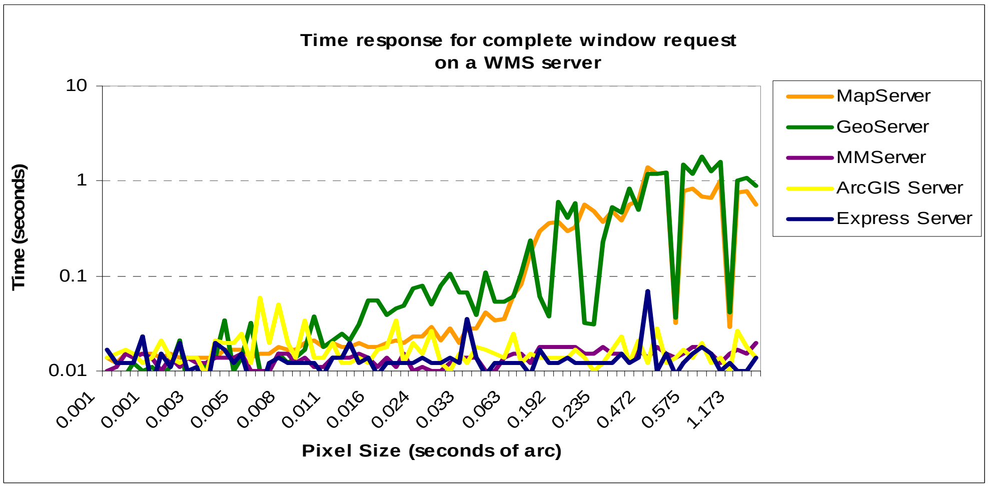


Figure 9. Regular WMS for full image requests.

Figure 9 shows that a full WMS viewport is the fastest for all servers, particularly for Express Server, ArcGIS Server and MiraMon Map Server, probably because the server only does the work once even if the volume of information to deliver is bigger. When tiles are used, requesting all tiles sequentially results in the slowest solution for all servers; however, limiting the number of concurrent requests to 4 improves the response time significantly. This is the best performance situation for MiraMon Map Server and GeoServer. After seeing this, it is easy to understand why many web browsers limit the number of simultaneous requests to a relatively small number (depending on the product and the version). Out of the concurrent tile request situations, this is the best tile solution for MapServer, Express Server and ArcGIS Server. Determining the optimum semi-concurrence number for each server will be the focus of a future work.

The tile products tested (TileCache and GeoWebCache) provide a way of pre-rendering tiles or saving tiles that are generated on the fly for further use. The main drawback is the generation time, but this can be partially overcome by *metatiling* strategies. Both TileCache and GeoWebCache support metatiling. Instead of generating each tile individually, a *metatile* is generated, creating a single large map image that can be divided into a number of tiles. Figure 10 shows that for all servers, analyzing a 512 x 512 image requires more time, but much less time than that required for analyzing four 256 x 256 images. This is because generating a metatile involves accessing source data only once instead of four times for a set of four tiles

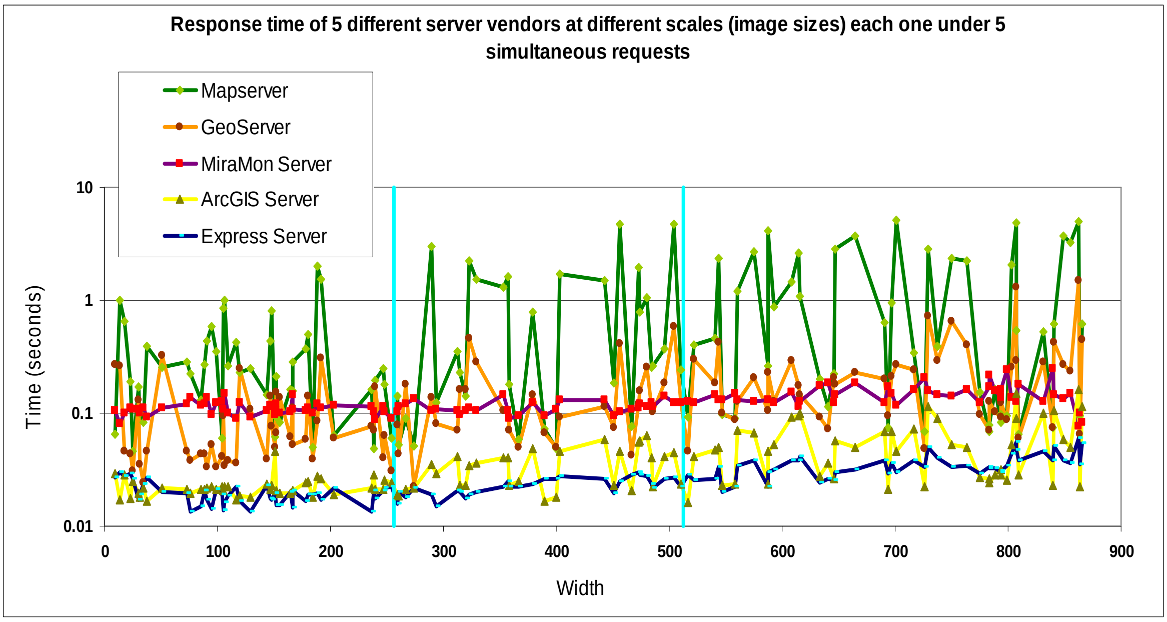


Figure 10. Response time for image size in concurrent requests to 5 single servers. Two marks at 256 and 512 pixel sizes have been added to facilitate the comparison between the response time in these two common pixel sizes.

VI. CONCLUSIONS

The speed tests described in this paper are a practical demonstration of the suitability of certain servers and service configurations in certain domains in which the reliability of the services under high stress conditions is imperative. This document summarizes and quantifies the results of our speed tests and determines which servers are faster under the minimum configuration.

All the analyzed servers have slower performances when the number of simultaneous clients is increased. A cluster server can be used to solve this situation: a group of computers is able to respond at the same time to different clients, assigning each client to a different computer in the group.

The results show that WMS servers do not perform well if clients using tile strategies are used over servers that are not optimized for tile response. MapServer and GeoServer with minimum data configuration do not require a data preparation process; however, they do not perform as well as other services that require indexing methods like MiraMon Map Server or Express Server. MapServer (based on C++ code) performs better than GeoServer (based on Java code) under single client requests, but GeoServer is surprisingly faster under concurrent simultaneous requests. The fastest WMS server is Express Server which works with MsSID or JPEG2000 compressed images that are 5% of the original size. The fastest tile server in the three cases assessed (concurrent, semi-concurrent and sequential requests) is GeoWebCache built over GeoServer.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Government, by FEDER, and by the Catalan Government, under Grants TSI2006-14005-C02-02, TIN2009-14426-C02-02 and GRUMETS 2009 SGR 1511, and also by the European Commission through the FP7-242390-GEO-PICTURES project (FP7-SPACE-2009-1). Xavier Pons is the recipient of an *ICREA Acadèmia* Excellence in Research grant.

REFERENCES

[1] Altan, O., Backhaus R., Boccardo, P., and Zlatanova, S. (2010), Geoinformation for Disaster and Risk Management. Examples and Best Practices, Copenhagen, Denmark. ISBN 978-87-90907-88-4.

[2] De la Beaujardiere, J. (2004) – OGC Web Map Service (WMS) Interface. Ver.1.3.0, OGC 03-109r1. Web: http://www.opengeospatial.org/standards/wms, last access March 1st 2011.

[3] ESRI (2010), Image Server & ArcGIS Server, 9.3. 380 New York St., Redlands, CA 92373-81000 USA.

[4] GeoServer (2010) user manual version 1.7, Web: http://docs.geoserver.org/, last access March 1st 2011.

[5] Google Crisis Response (2010), Haiti Earthquake – GeoEye Satellite Imagery Download, Web: http://www.google.com/relief/haitiearthquake/geoeye.html, last access March 1st 2011.

[6] LizardTech (2010), LizardTech Inc y Lizardtech España SL. GeoExpress compressor. version 8 & Express Server™ version 6.1.

[7] Map Server Team, (2010), MapServer Release 5.6.5 documentation, Open source Web Mapping, Web: http://mapserver.org/tutorial/index.html, last access May 24th 2011.

[8] Marin, L., (2008) Setting up TileCache on IIS, in a blog at WorldPress.com Web: http://viswaug.wordpress.com/2008/02/03/setting-up-tilecache-on-iis/, last access March 1st 2011.

[9] Masó J., Pomakis K., and Julià N. (2010a) – OGC Web Map Tile Service (WMTS). Implementation Standard. Ver 1.0, OGC 07-057r7 Web: http://www.opengeospatial.org/standards/wmts, last access March 1st 2011.

[10] Masó J., Zabala A., and Pons X. (2010b) Combining JPEG2000 Compressed Formats and OGC Standards for Fast and Easy

Dissemination of Large Satellite Data, Italian Journal of Remote Sensing - 2010, 42 (3): 101-114

[11] Masó J, Díaz, P., and Pons, X. (2011) Performance of standardized web map servers for remote sensing imagery. In: Proceedings of "Data Flow: From Space to Earth. Applications and interoperability Conference" (21-23th March 2011, Venice).Corila - Consorzio per la Gestione del Centro di Coordinamento delle Attività di Ricerca Inerenti il Sistema Lagunare di Venezia. pp.83-83 ISBN:9788889405154.

[12] Matt Mills, NASA World Wind Tile Structure. Web: http://www.ceteranet.com/nww-tile-struct.pdf, last access March 1st 2011.

[13] MetaCarta Labs, TileCache Contributors (2006-2010), TileCache Web: http://tilecache.org/, last access May 24th 2011.

[14] Nie Y., Xu, H., and Liu H., (2011), The Design and Implementation of Tile Map Service, in Advanced Materials Research, Vol. 159 (2011) pp 714-719, Trans Tech Publications, Switzerland.

[15] OpenGeo (2011), GeoWebCache User Manual, Web: http://geowebcache.org/docs/current/, last access May 24th 2011.

[16] OSGeo (2011), Tile Map Service Specification, OSGeo Wiki. Web: http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification, last access March 1st 2011.

[17] OSGeo, OpenLayers wiki, Using Custom Tile Sources, Google-like Tile Layer Support, Web: http://trac.osgeo.org/openlayers/wiki/UsingCustomTiles, last access March 1st 2011.

[18] Pons, X., (2004) MiraMon. Geographic Information Systems and Remote Sensing software. v. 4, Center for Ecological Research and Forestry Applications, CREAF, Bellaterra. 286 p. ISBN: 84-931323-4-9.

[19] GEOSS, The Global Earth Observation System of Systems. Web: http://www.earthobservations.org/