

Enhancing Job Scheduling of an Atmospheric Intensive Data Application

Olivier Terzo, Lorenzo Mossucca, Klodiana Goga, Pietro Ruiu, Giuseppe Caragnano

Infrastructure and Systems for Advanced Computing (IS4AC)

Istituto Superiore Mario Boella (ISMB)

Turin, Italy

(terzo, mossucca, goga, ruiu, caragnano)@ismb.it

Abstract—Nowadays, e-Science applications involve great deal of data to have more accurate analysis. One of its application domains is the Radio Occultation which manages satellite data. Grid Processing Management is a physical infrastructure geographically distributed based on Grid Computing, that is implemented for the overall processing Radio Occultation analysis. After a brief description of algorithms adopted to characterize atmospheric profiles, the paper presents an improvement of job scheduling in order to decrease processing time and optimize resource utilization. Extension of grid computing capacity is implemented by virtual machines in existing physical Grid in order to satisfy temporary job requests. Also scheduling plays an important role in the infrastructure that is handled by a couple of schedulers which are developed to manage data automatically.

Keywords-virtualization; grid computing; job scheduling; scalability; radio occultation; distributed environment.

I. INTRODUCTION

Italian Space Agency (ASI) [1] developed a GPS receiver devoted to Radio Occultation (RO). The space-based GPS limb sounding, conventionally known as GPS Radio Occultation, is a remote sensing technique for the profiling of atmospheric parameters: refractivity, pressure, temperature, humidity and electron density, see [2], [3]. It is based on the inversion of GPS signals collected by an ad hoc receiver placed on-board a Low Earth Orbit (LEO) platform, when the transmitter rises or sets beyond the Earth's limb. The relative movement of both satellites allows a quasi vertical atmospheric scan of the signal trajectory and the profiles extracted are characterized by high vertical resolution and high accuracy. The RO technique is applied for meteorological purposes (data collected by one LEO receiver placed at 700 km altitude produce $300 \div 400$ profiles per day, worldwide distributed) since such observations can easily be assimilated into Numerical Weather Prediction models. Furthermore, it is also very useful for climatological purposes, for gravity wave observations and for Space Weather applications. This will cause the phase of the signal to be delayed. Moreover, the bent Geometric Optics trajectories followed by the signal during an entire occultation event will span the entire atmosphere in the vertical direction. As a consequence, through the inversion of the phase delay measurements, the refractivity related to each trajectory perigee can be evaluated, and a vertical profile can be identified. From

refractivity, and adopting variational techniques, temperature and water vapor profiles can also be inferred. Given the characteristics of global coverage, good accuracy and high vertical resolution, products derived using such a technique are operationally used as input to weather forecasting model tools, and could also be harnessed in monitoring climate changes. ROSA-ROSSA (Radio Occultation Sounder for Atmosphere - Research Operational Satellite and Software Activities) is integrated in the operational ROSA Ground Segment it is operating at the ASI Space Geodesy Center, in Matera, Italy and at the Indian National Remote Sensing Agency [4], in Sriharikota, India. In this framework, Italian Space Agency has funded the development of the operational RO Ground Segment, which include the ROSA-ROSSA software. Partners of this project are several Italian universities, research centers and one industrial actor, which are responsible for the development and the integration of the various software modules defining the ROSA-ROSSA software: Istituto Superiore Mario Boella (Turin), Polytechnic of Turin, "Centro Interdipartimentale di Studi e Attività Spaziali" of Padua, University "La Sapienza" of Rome, University of Camerino, "International Center of Theoretical Physics" of Trieste, "Institute for Complex System" of Florence and "Consorzio per l'Informatica e la Telematica" of Matera.

The paper is structured as follows: Section 2 explains related work, motivation is shown in Section 3, Section 4 presents algorithms for Radio Occultation, Section 5 describes Grid architecture and virtual environment adopted to enhance capacity, Section 6 provides a brief overview about scheduling approach and its issue, Section 7 depicts some performance analysis and last Section draws conclusions and future work.

II. RELATED WORK

The existent system is managed by an integrated software, called Grid Processing Management (GPM), devoted to handle and process data of the OCEANSAT-2 on board sensor. This architecture consists of the following components: physical worker nodes, repository, relational database, scheduler, agents and applications. The observed data, once acquired by the receiving ground station, are processed to produce refractivity, temperature and humidity profiles. The Radio Occultation events data processing consist of seven

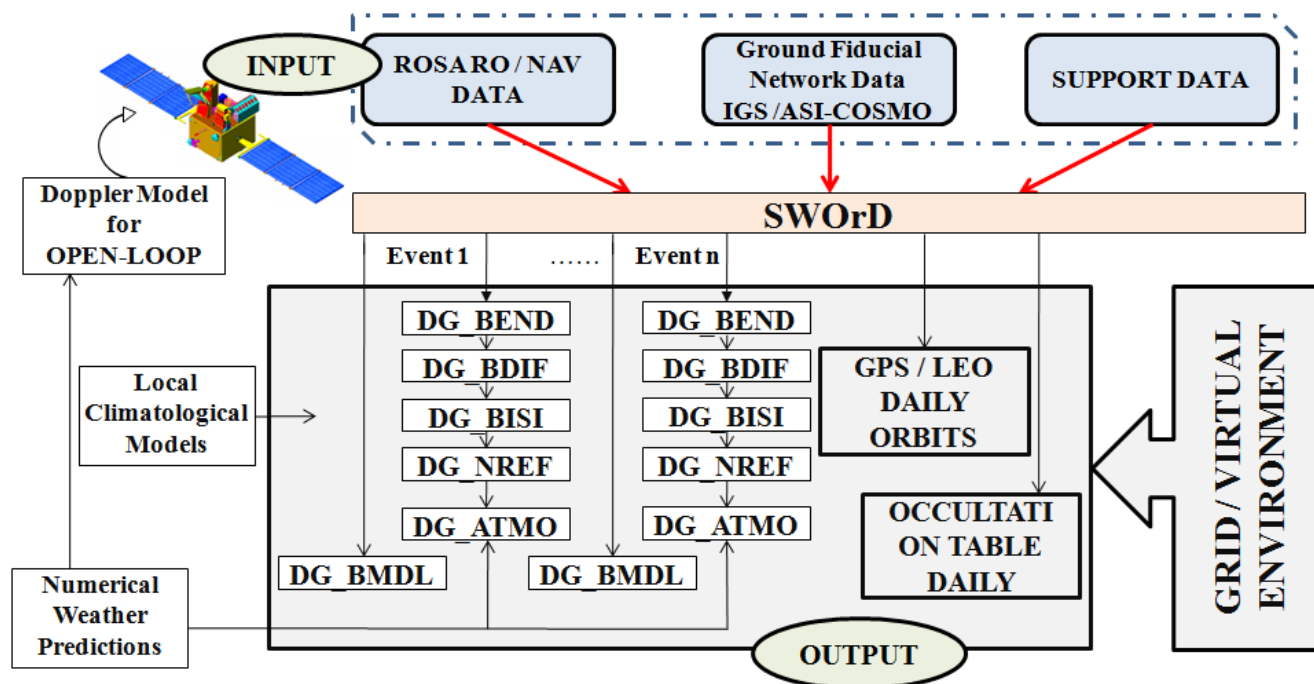


Figure 1. Data Generators Workflow.

main steps, named Data Generators (DGs). Figure 1 depicts the whole list of DGs that must be executed sequentially. Input and output data are daily composed of about 256 occultation events to be processed sequentially. For further details see [5]. In this context, where one needs to elaborate an enormous amount of data, using a grid architecture, there is already a great saving of time. In some cases the system fills up, when all worker nodes are elaborating data, increasing execution time. A solution to solve this problem can use a dynamically scalable system with virtual machines. The proposed architecture thus consists of a virtualized environment, which adds to the grid virtual nodes on demand, in order to increase the computational power and to solve load temporary peaks. In addition it allows to create multiple virtual nodes on the same machine optimizing physical resources, reducing energy consumption and decreasing maintenance costs. Virtualized systems help to improve infrastructure management, allowing the use of image template to create virtual nodes in a short time, speeding up the integration of new nodes on the grid and, therefore, improving reactivity and scalability of the infrastructure.

III. MOTIVATION

The project aims to create a flexible architecture in order to manage the Radio Occultation data and to reduce their processing time. System guarantees the entire processing chain automatically that consists of seven DGs executed sequentially as explained before. During a learning phase,

we evaluated that for each day, the events number to process are about 250, on a single machine the elaboration time for the entire chain processing, is approximatively 40 hours. The idea of using a distributed environment arose from the need to reduce this processing time because these make it difficult to get the results. The GPM has been developed with the goal of simplifying this task, by providing implementations of various core services provided by Globus Toolkit [6] and deemed essential for high performance distributed computing. Furthermore, it allows engineers and physicists involved to the project to have a common tool and infrastructure to process and share data, independently from the university in which they are. The chain process is composed of seven jobs developed by complex algorithms that involve a set of languages as Fortran, MatLab, C++, Mathematica, Java and Perl. Based on the limitations of software due to several programming languages involved in DGs algorithms, we decided to develop an ad-hoc job management scheduler stressing the importance of system scalability for grid infrastructure. A way to provide flexibility and scalability to the system is the implementation of a multi agents solution. The scheduler never gets information requiring the status to each nodes in fact on each node are installed two types of agents: Job and System agent. The first one is used to monitor the behavior of CPU, RAM and swap of nodes during the DG execution and it handles sending these information to master node. The second one, System agent, is used to monitor availability of each service on the node and periodically sends to the master node its own status: if all services

are available the node is ready to receive a job. The main advantage is that scheduler retrieves a pre list of available nodes ready for execution only with a simple query to the database.

IV. RADIO OCCULTATION ALGORITHMS

ROSA-ROSSA software implements state of the art RO algorithms which are subdivided into seven different DGs executed in sequential mode [7], [8]. Starting from ROSA engineered data (or raw data observed by other RO payloads made available to the scientific community) coming from the ROSA on-board OCEANSAT-2 platform observations, from the ground GPS network (i.e., International GNSS Service network) and from other support data, the ROSA-ROSSA is able to produce data at higher levels, using a data processing chain defined by the following DGs:

- SWOrD;
- DG_BEND;
- DG_BDIF;
- DG_BISI;
- DG_NREF;
- DG_ATMO;
- DG_BMDL.

A. DGs Description

SWOrD is a software module that fully supports the orbit determination, orbit prediction, and which implements data generation activities connected with the ROSA sensor on-board OCEANSAT-2. Input data for SWOrD are ROSA GPS navigation and Radio Occultation observations, ground GPS network data and other support data. It generates the following output data:

- estimated rapid orbits and predicted orbits for the GPS constellation in Conventional Terrestrial Reference frame;
- estimated rapid orbits and predicted orbits for the OCEANSAT-2 platform in Conventional Terrestrial and Celestial Reference frame;
- 50 Hz closed-loop and 100 Hz Open-Loop excess phases and signal amplitudes data for each single occultation event;
- tables showing estimated and predicted (up to 6 hours in advance) occultation.

DG_BMDL predicts a bending angle and impact parameter profile usable as input in the ROSA on-board software Excess Doppler prediction module for open-loop tracking (see Figure 2). For each "predicted" occultation event, latitude and longitude of the geometrical tangent points (the nearest point of each trajectory to the Earth's surface, evaluated through predicted orbits) is used to compute bending angle and impact parameter profile from interpolated numerical weather prediction models (bending angle and impact parameter are geometrical parameter univocally identifying each trajectory followed by the RO

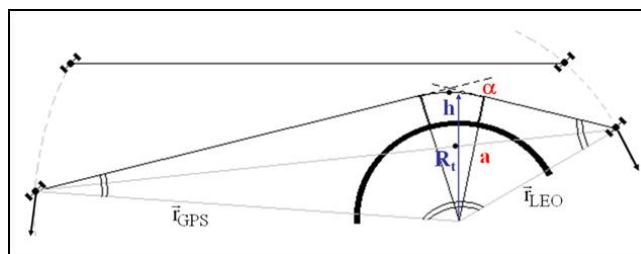


Figure 2. Radio Occultation Geometry.

signal). Inputs for DG_BMDL are predicted GPS and LEO orbits, respectively, and Predicted Occultation Tables, together with European Centre for Medium-Range Weather Forecasts (ECMWF) [9] world forecasts for the synoptic times valid for the future observed occultation event.

DG_BEND provides raw bending angle and impact parameter profiles $\alpha(a)$ computed on GPS occulted signals on both GPS frequencies L_1 and L_2 , by using a Wave Optics approach below a certain altitude. Above that altitude threshold, standard Geometrical Optics algorithms are applied. Inputs for DG_BEND are data L_1 and L_2 Excess Phases and signal amplitudes.

DG_BDIF provides (for each event) a bending angle and impact parameter profile, on which the ionospheric effects have been compensated for. This DG processes both L_1 and L_2 bending angle and impact parameters profiles given as input, in order to minimize the first order ionospheric dispersive effects. Outputs for DG_BDIF are bending angle and impact parameter iono-free profiles.

DG_BISI provides profiles of bending angle versus impact parameter optimized in the stratosphere above 40 km. In the ROSA-ROSSA, data coming from a Numerical Weather Prediction Model are used in place of climatological data for implementing the statistical optimization procedure necessary to reduce the high noise level left to the signal after ionospheric first order compensation previously applied by DG_BDIF.

DG_NREF provides (for each event) the refractivity profile and dry air temperature and pressure profiles. This DG is able to process iono-free and properly initialized bending angle and impact parameter profiles in order to compute the corresponding dry air "quasi" vertical atmospheric profiles.

DG_ATMO allows to evaluate the temperature and the water vapor profiles using forecasts or analysis obtained by numerical weather prediction. This DG receives on input from DG_NREF data files and produces on output data files, which contain the total temperature and total pressure

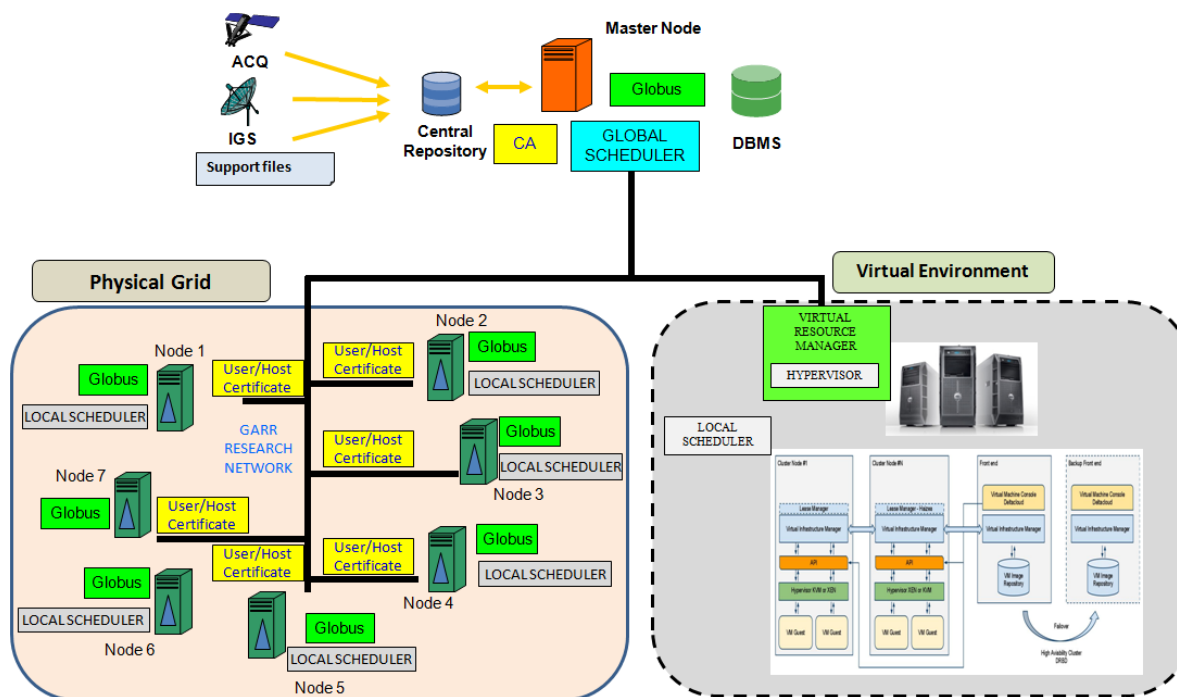


Figure 3. System Architecture.

profiles in terms of wet and dry components.

V. ARCHITECTURE DESIGN

The Grid Processing Management is an integrated system devoted to handle and process RO data of the OCEANSAT-2 ROSA on board sensor. The management software is developed in Java technology through the Java Commodity Grid (CoG) Kits. These allow to develop Grid application and administer Grids from a higher-level framework. The kits allow for easy and rapid application development and are used within the services of Globus Toolkit.

A. Grid Processing Management

The GPM is composed of the following subsystems (see Figure 3): middleware, central repository, relational database, scheduler, agents and DGs applications. The general purpose of our project is: sharing the computational resources, transferring a great amount of files and submitting jobs from several different organizations of the scientific community. Once input files stored, they are processed in an automatic way without any user interaction. The pool of resources consists of ten nodes: one Master Node and nine Worker Nodes, located geographically in Italy, each of them is equipped with a dual core processor, 2 GB of RAM and runs Linux x64_86 (Ubuntu distro). Master Node is responsible for the Data Base Management System, Certificate Authority, resource and job monitoring and job scheduling. Instead Worker Nodes contains grid and software tools used by jobs running which are received from Master Node.

The Globus Toolkit (GT) is a popular grid middleware [6], [10]. It provides a set of tools to create a grid infrastructure, covering security measures, resource location, resource management, communications which support the development of applications for high performance distributed computing environments, or computational grids [11], [12].

The main services included in GT are:

- Globus Resource Allocation Manager (GRAM): it converts a job request for resources into commands;
- Grid Security Infrastructure (GSI): for authentication of users and determines their access policies;
- Monitoring and Discovery Service (MDS): it collects information about resources such as processing capacity, bandwidth capacity, type of storage;
- Grid Resource Information Service (GRIS): it queries resources for their configuration, capabilities, and status;
- Grid Index Information Service (GIIS): coordinates arbitrary GRIS services;
- Grid File Transfer Protocol (GridFTP): provides a high-performance, secure, and robust data transfer mechanism.

The main reason for the choice of the GT is the availability under an open-source licensing agreement, which allow to use it freely and to improve the software. Grids need to support a wide variety of applications created according to different programming paradigms. Rather than providing a uniform programming model for grid applications, GT has

an object-oriented approach, providing a bag of services so that developers can choose the services that best meet their needs. These can also be introduced one at a time.

B. Extending Grid Capacity through Virtual Environment

Virtualization is a technology that allows running several concurrent operating system instances inside a single physical machine, reducing the hardware costs and improving the overall productivity by allowing users work on it simultaneously. The hypervisor, the fundamental component of a virtualized system, provides infrastructure support exploiting lower-level hardware resources in order to create multiple independent Virtual Machines (VM), isolated from each other. This virtualized layer, called also Virtual Machine Monitor (VMM), sits on top of the hardware and below the operating system. The hypervisor can control (create, shutdown, suspend) each VM that is running on top of the host machine. Multiple instances of different operating systems may share the virtualized hardware resources. The hypervisor is so named because it is conceptually one level higher than a supervisory program. A supervisory program or supervisor, also called kernel, is usually part of an operating system, that controls the execution of other routines and regulates work scheduling, input/output operations, error actions, and similar functions and regulates the flow of work in a data processing system (see Figure 4). Virtualization allows to gain significant benefits from the economic and the optimization of resources point of view [13]. Besides these, other noteworthy benefits are:

- security, stability and isolation: it is possible to run services in a virtual environment totally independent from each other;
- environmental impact reduction: optimization of resources implies reduction of power consumption and cooling;
- administration and management simplification: due to the common virtualization layer and the adoption of snapshots (installation and configuration);
- disaster recovery: VM can be started up in few minutes and can be cloned and distributed in different locations;
- high reliability and load balancing improvement: thanks to snapshots and live migration features.

Another aspect of virtualization is the adaptability: in fact it allows resource allocation to virtual hardware easily and quickly. We can maintain old servers with obsolete operating systems that cannot be moved to new servers as these OS would not be supported. In virtualized environments it is possible to run legacy systems allowing IT managers to get rid of old hardware no longer supported, and more prone to failure. In several cases it is appropriate to use virtualization to create test environments. It frequently happens that production systems need to be changed without knowledge about consequences, i.e., installing an operating

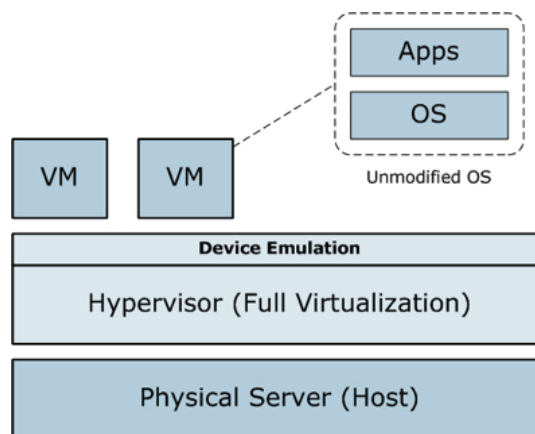


Figure 4. Full Virtualization.

system upgrade or a particular service pack is not a risk-free. Virtualization allows immediate replication of virtual machines in order to run all necessary tests.

C. Automatic chain

Our software allows to run the chain automatically and it is composed of two schedulers. The first runs on the Master Node, called Global Scheduler, checks for files ready for execution and sends them to Worker Nodes, according to well defined scheduling policies. The other one, called Local Scheduler, listens on the Worker Nodes, and when an input file is received, it is processed and output is returned to the Master Node [14], [20]. In Figure 5, data flow is depicted, the first transaction takes place on Master Node: it receives the files directly from the satellite and performs the first step of the chain, i.e., SWOrD, generating about 256 files that are placed in the input folder of the next step, DG_BEND. When there are files in the folder DG_BEND, the Global Scheduler checks available nodes by querying the database, and sends files to them. Global Scheduler takes care of automated scheduling of any input file. It uses all machines belonging to the Grid to distribute work load and to provide a backup system for all critical tasks within the system. The choice of how to share the file to run is based on 2 sets of scheduling policies, one concerning the available nodes and one derived from an analysis of the file to run. An Agent installed on each node, is used to monitor the availability of each service on the node. Periodically, it sends the general status of the node to the database on Master Node: if all services are active the node is in condition to receive a job. For the selection of Worker Nodes available and ready to run, the Global Scheduler checks on the database directly instead of querying each machine. When the Worker Node becomes aware of a file in its folder, the processing procedure starts. This will generate an output file that will be sent to Master Node in the next step folder, i.e., DG_BDIF. This procedure is performed for every steps of the chain, the operation is as

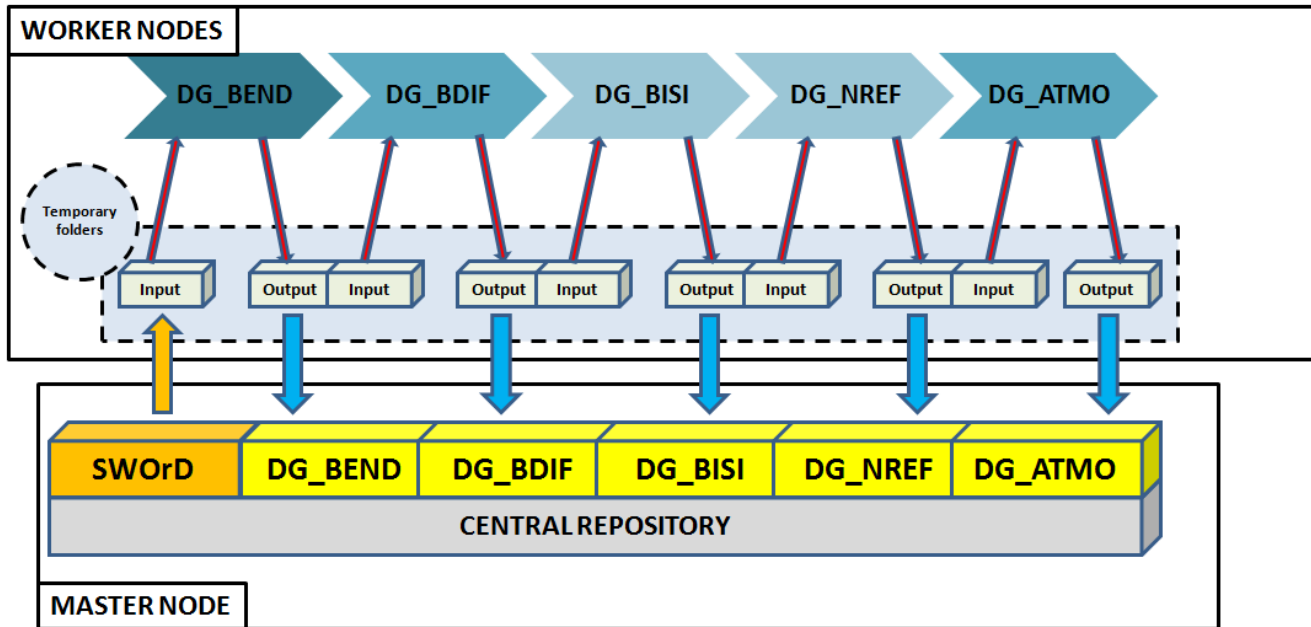


Figure 5. Chain Context.

follows: from SWOrD, the DG n-1 generates the output file that will be the input files of DG n, and so on. On Worker Nodes, each execution is performed in a temporary folder, so that, in case of error, it could be possible to identify the type of error made and then to reprocess the file. Two types of errors can occur: the first for lack of data in the file due to the satellite reception, the second for network failures or node crash. Only in the last case it is worth recover the process, and it is enough reprocess the input file. Furthermore, each process has a timeout, if within a fixed time processing has not been completed, the process is terminated in anyway. An important component of this architecture is the database, which allows to monitor any action of the Grid. Each transaction is stored on the database such as selected DG, start and end end time, input and output files, execution node and can contain type of error generated. The database also contains information on node status and if are available to receive the file to run, this allows to understand whether there are temporary network problems, so if the node is not able to receive the job.

VI. SCHEDULING APPROACH

The Grid scheduler selection is an important decision and significantly affects cluster utilization, availability, and intelligence. The most widely used are Portable Batch System, Torque scheduler, Maui Scheduler, Moab Workload Manager and Oracle Grid Engine, but they provide poor utilization of cluster's resources. These schedulers are enough closed and not are not easy to customize, in our system each machine is configured with different softwares and that means DG can not be executed on any Worker Node. This implies a

most complicated job dispatching procedure that need of a high level of customization. The assignment of node for a new job execution is based not only on resource availability but also software needed for execution. To understand our choice to make a scheduler ad hoc in the following we want to give a brief overview of existing schedulers.

Portable Batch System (PBS)[15] is a networked subsystem for submitting, monitoring, and controlling a workload of batch jobs on one or more systems. Its main task is to allocate computational tasks, i.e., batch jobs, among the available computing resources. PBS is supported as a job scheduler mechanism by several meta schedulers including Moab and GRAM (Grid Resource Allocation Manager). With PBS it can specify the tasks to be executed; the system takes care of running these tasks and returns its results. If all computers are busy, then PBS holds your work and runs it when the resources are available. With PBS it is possible to create a batch job and then submit it. A batch job consists of a file containing the set of commands to run. It also contains directives which specify the attributes of the job, and resource requirements (e.g., number of processors and CPU time) that the job needs. Once PBS job is created, you can reuse it or modify it for subsequent runs.

Torque Resource Manager [16] provides control over batch jobs and distributed computing resources. Its name stands for Terascale Open-Source Resource and QUEUE Manager. It is an open-source product based on the original PBS project and incorporates significant advances in the areas of scalability, reliability, fault tolerance, features extensions and functionality and is currently in use at tens of thousands

of leading government, academic, and commercial sites throughout the world. Torque can integrate with Moab Workload Manager to improve overall utilization, scheduling and administration on a cluster. The Torque Resource Manager is a distributed resource manager providing control over batch jobs and distributed compute nodes.

Maui Cluster Scheduler [17] is an open source job scheduler for clusters and supercomputers. It is a configurable tool capable of supporting an array of scheduling policies, dynamic priorities, extensive reservations, and fairshare capabilities. All of the capabilities found in Maui are also found in Moab, while Moab has added features including virtual private clusters, basic trigger support, graphical administration tools, and a Web-based user portal.

Moab [18] is a multi-dimensional policy-based workload management system that accelerates and automates the scheduling, managing, monitoring, and reporting of HPC (High Performance Computing) workloads on massive scale, multi-technology installations. The Moab accelerates both the decisions and orchestration of workload across the ideal combination of diverse resources, including specialized resources as GPGPUs. The speed and accuracy of the decisions and scheduling automation optimizes workload throughput and resource utilization so more work is accomplished in less time with existing resources to control costs and increase the value out of HPC investments.

Oracle Grid Engine [19], previously known as Sun Grid Engine (SGE), is an open source and free batch-queuing system, developed and supported by Sun Microsystems. SGE is typically used on a computer farm or high performance computing cluster and is responsible for accepting, scheduling, dispatching, and managing the remote and distributed execution of large numbers of standalone, parallel or interactive user jobs. It also manages and schedules the allocation of distributed resources such as processors, memory, disk space, and software licenses.

A. Job Management

The scheduling process takes place on the Master Node through the Global Scheduler (GS) [21]. This scheduler, developed in Java, is running on each folder dedicated to the DG, it waits for new events. Every time it takes notice of a new file to be executed, GS assigns and delivers it to the selected node according to predefined policies. The policies are split in the following steps:

- resources discovery is directly related to the information sent by the agents in order to return a pre list of available nodes;
- software discovery, by DG chose, selects the node that contains the software needed to run;
- status discovery devoted to check which nodes are free for execution (not running);
- hardware control: sorts the pre list based on computer power;

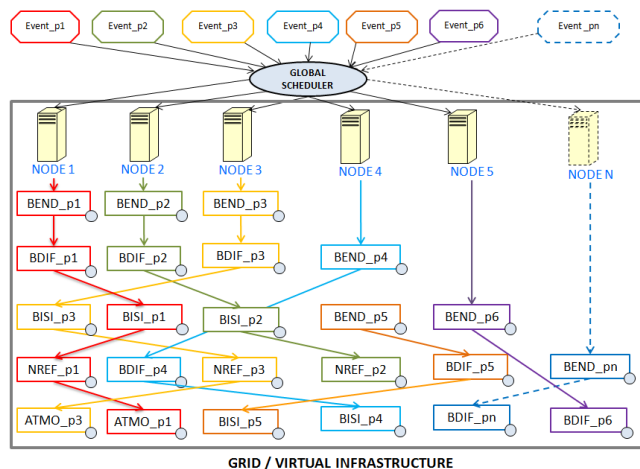


Figure 6. Dispatching Jobs.

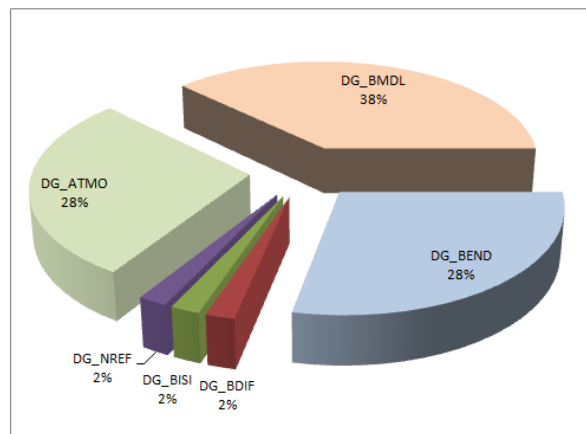


Figure 7. DG Percentage of Total Processing Time for a Single Event.

- queue control: at equality of power, it assigns job to node with less jobs in waiting.

VII. IMPROVING PERFORMANCES

Performance analysis have been executed for any DG, in order to improve scheduling policies. Since the data distribution is not homogeneous, this means that every hour SWOrD generates about 9 files which go in the folder DG_BEND and instead only 1 in folder DG_BMDL. Consequently for a day we have about 216 files for DG_BEND and 24 files for DG_BMDL. Figure 7 and Figure 8 represent the comparison for hourly and daily events, this allowed to assign a weight to each DG in order to calculate the amount of time when the CPU is busy.

The processing time of SWOrD step (about 72 minutes) is not included in these graphs because is executed only on Master Node therefore we can considered it outside from the grid. In order to compare performances between single machine and grid/virtual environment, we calculated processing time through the following Equations, varying

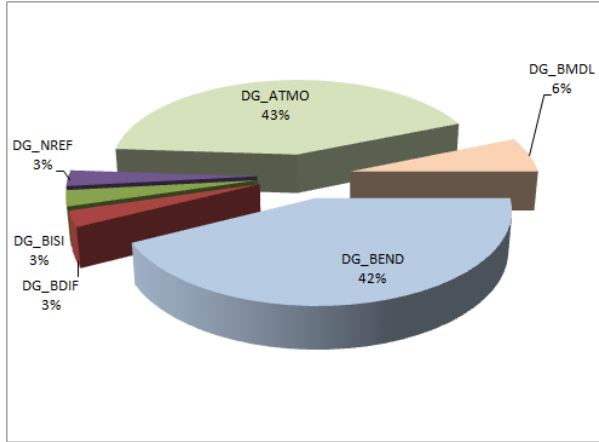


Figure 8. DG Percentage of Total Processing Time for Daily Events.

nodes number. The Eq. 1 is referred to elaboration step by step sequentially on a single machine then $N = 1$.

$$T_{N=1} = T_e * \eta \quad (1)$$

While Eq.2 represents Grid/Virtual Environment that means $N > 1$. In this case, it is important to consider time for file transferring.

$$T_{N>1} = \frac{1}{N} \sum_{i=1}^{\eta} (T_{ei} + \beta) \quad (2)$$

Where: $T_{N=1}$ = Total Process Time on Single Machine, $T_{N>1}$ = Total Process Time on Grid/Virtual Environment, T_e = Event Process Time, T_{ei} = Event i Process Time, η = Number of RO Events, N = Grid Nodes Number, β = File Transfer Time.

Execution time trend is estimated in Figure 9, when the number of nodes and events is increased. When only a single machine is available, the total execution time for a set of daily files is 1752 minutes (about 29 hours), instead increasing the number of nodes, the execution decrease further, just note that with 2 nodes is 912 (about 15 hours). An important point when a single event is processed is that there is no gain time in grid environment respect to single machine execution; rather time is higher because we must consider the transfer time; it has a sizeable gain time only when there are a set of files to process.

Certainly, the benefits of the grid is ensure elaboration the overall chain in less time, instead, in distributed system where Worker Nodes are geographically located, it can have disadvantage in the network layer, in case of network failures or slow connections, to overcome this problem only internal nodes are always available for elaboration.

In Figure 10 we evaluated the elaboration time of each Data Generators executed on two types of nodes: physical and virtualized node. For DG_BDIF, DG_BISI, DG_NREF

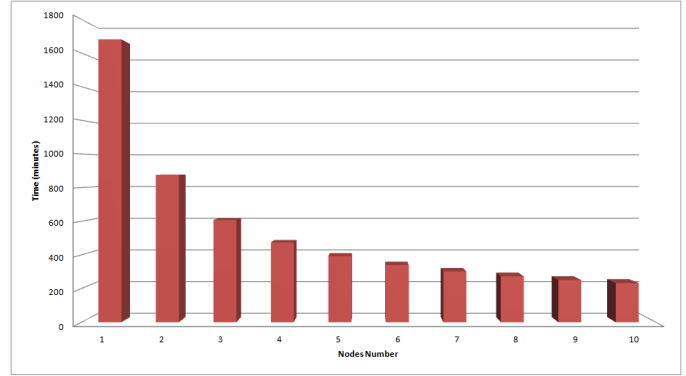


Figure 9. Estimated Processing Time for Daily Events (about 250 events)

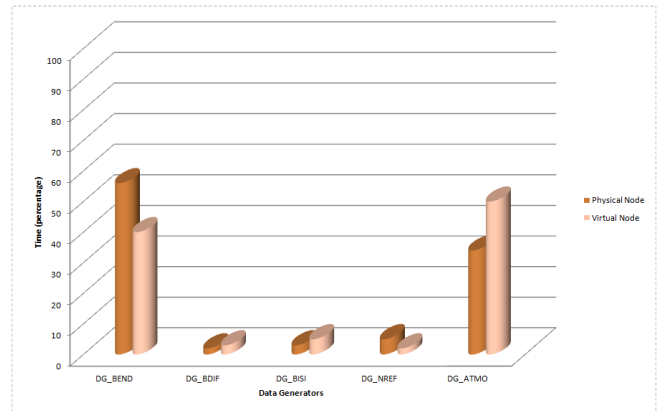


Figure 10. Comparison Physical Node vs Virtual Node.

we can notice we have no elaboration time difference. Instead for DG_BEND and DG_ATMO we have a considerable time difference this can be due to different processor power (1.6 GHz for physical machines and 2.3 GHz for virtual machines). The server used is equipped with a dual-core Intel Xeon (4 CPU), 8 GB of RAM and 130 GB of storage. The virtual machines reside entirely on this server and therefore they share the resources (RAM, CPU, disk); each virtual node has 2 GB of RAM and 2 dedicated CPUs. They are configured exactly like a physical node of the grid with the same softwares and monitoring tools. It was decided to use Para Virtualized systems since it was shown that (in terms of network and I/O), they have better performances than the Full Virtualized one [22].

Para Virtualization Machine allows the operating system to be aware that it is running on a hypervisor instead of base hardware. The operating system must be modified to accommodate the unique situation of running on a hypervisor instead of basic hardware. The main advantage of this approach is the execution speed, always faster than HVM and Full Virtualization approach. The Xen hypervisor runs directly on the hardware and becomes the interface for all hardware requests such as CPU, I/O, and disk for the guest

operating systems. XEN is responsible for CPU scheduling and memory partitioning of the various virtual machines running on the hardware device. The hypervisor abstracts the hardware for the virtual machines and controls the execution of virtual machines as they share the common processing environment. It has no knowledge of networking, external storage devices, video, or any other common I/O functions found on a computing system.

VIII. CONCLUSION AND FUTURE WORK

The ROSA-ROSSA software implements Radio Occultation technique, which was born for the first time on a Grid Computing infrastructure, called Grid Processing Management and in a second moment has been implemented a virtual environment to improve computing power and resources optimization. The project aims to be an example of application where users can use Grid Computing. In frameworks such as Radio Occultation, where the amount of data to be processed is significant, the use of a distributed architecture as the grid can be the optimal choice. We wanted to stress on a way to distribute jobs to nodes for execution in automatic way without any human interaction through a Local and a Global Scheduler. In order to increase computing capacity several virtual machines are added to the existing Grid infrastructure. As future work we are planning a further extension of the proposed architecture to clusters available across the European Grid Infrastructure (EGI) it provides access to high-throughput computing resources across Europe using grid computing techniques, furthermore we are studying a solution for Amazon Elastic Compute Cloud (Amazon EC2), that is a web service that provides resizable compute capacity in the cloud.

IX. ACKNOWLEDGMENT

The authors are grateful to the Italian Space Agency for supporting this project within the contract I/006/07/0 and to all ROSA-ROSSA partners for their contributions.

REFERENCES

- [1] Italian Space Agency(ASI), Available at: <http://www.asi.it/>, 2012
- [2] Melbourne, W.G., Davis, E.S., Duncan, C.B., Hajj, G.A., Hardy, K.R., Kursinski, E.R., Meehan, T.K., Young, L.E. and Yunck T.P.: *The application of spaceborne GPS to atmospheric limb sounding and global change monitoring*, JPL Publication, (1994), pp. 18-94
- [3] Kursinski, E.R., Hajj, G.A., Schofield J.T., Linfield R.P., and Hardy K.R.: *Observing Earths atmosphere with radio occultation measurements using the Global Positioning System*, Journal of Geophysical Research 102(D19), (1997), pp. 23.429-23.465
- [4] Indian Space Research Organization(ISRO), Available at: <http://www.isro.org/>, 2012
- [5] Mossucca L., Terzo O., Molinaro M., Perona G., Cucca M., Notarpietro R.: *Preliminary results for atmospheric remote sensing data processing through Grid Computing*, The 2010 International Conference on High Performance Computing and Simulation, (2010), pp. 666-671
- [6] The Globus Alliance, Available at: <http://www.globus.org/>, 2012
- [7] Wickert J., Schmidt T., Beyerle G., Knig R., Reigber C. and Jakowski N.: *The radio occultation experiment aboard CHAMP: Operational data analysis and validation of vertical atmospheric profiles*, Journal of the Meteorological Society of Japan 82(1B), (2004), pp.381-395
- [8] Luntama, J.P., Kirchengast, G., Borsche, M., Foelsche, U., Steiner, A., Healy, S., von Englern, A., O'Clery, E. and Marquardt, C.: *Prospects of the EPS GRAS mission for operational atmospheric applications*, Bulletin of the American Meteorological Society 89(12): (2008), pp. 1863
- [9] *European Centre for Medium-Range Weather Forecasts*, Available at: <http://www.ecmwf.int/>, 2012
- [10] The Globus Consortium, Available at: <http://www.globusconsortium.org/>, 2012
- [11] Foster, I. and Kesselman C.: *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Francisco, CA, (2003) pp. 38-63
- [12] Berman, F., Fox, G. and Hey A.: *Grid computing making the global infrastructure a reality*, Wiley, Chichester (2003), pp. 117-170
- [13] *The Future Of Cloud Computing, Expert Group Report*, European Commission, Information Society and Media, (2010), pp. 14-15
- [14] Dimitriadou, S. and Karatza, H.: *Job scheduling in a distributed system using backfilling with inaccurate runtime computation*, *International Conference on Complex, Intelligent and Software Intensive System*, Washington DC, USA, (2010), pp. 329-336
- [15] Portable Batch System, Available at: <http://www.pbsworks.com/SupportDocuments.aspx>, 2012
- [16] Torque Resource Manager, Available at: <http://www.adaptivecomputing.com/products/open-source/torque/>, 2012
- [17] Maui Cluster Scheduler, Available at: <http://www.clusterresources.com/products/maui-cluster-scheduler.php/>, 2012
- [18] Moab, Available at: <http://www.adaptivecomputing.com/products/hpc-products/moab-hpc-basic-edition/>, 2012
- [19] Oracle Grid Engine, Available at: <http://www.oracle.com/us/products/tools/oracle-grid-engine-075549.html>, 2012
- [20] Xhafa, F., Pllan, S. and Barolli, L., *Grid and P2P Middleware for Scientific Computing Systems*, The international conference on complex, intelligent and software intensive system, 2010, pp. 409-414

- [21] Leonid O., Rupak B., Hongzhang S. and Warren S.: *Job scheduling in a heterogeneous grid environment*, Lawrence Berkeley National Laboratory, (2004), Available at: <http://www.escholarship.org/uc/item/6659c4xj>
- [22] Kurowski, K., Nabrzyski, J.,A., Oleksiak, A. and Weglarz, J.: *Scheduling jobs on the grid multicriteria approach*, Computational Methods in Science and Technology 12(2), (2006), pp 123-138