Cloud-based Healthcare:
Towards a SLA Compliant Network Aware Solution for Medical Image Processing

Shane Hallett[1], Gerard Parr[1], Sally McClean[1], Aaron McConnell[1], Basim Majeed[2]
India-UK Advanced Technology Centre (IU-ATC) of Excellence in Next Generation Networks, Systems and Services
School of Computing and Information Engineering, University of Ulster[1], Coleraine, UK,
hallett-s@email.ulster.ac.uk, {gp.parr, si.mcclean, a.mcconnell}@ulster.ac.uk,
ETISALAT BT Innovation Center[2], Abu Dhabi, UAE, basim.majeed@bt.com

***Abstract*—Medical image processing in the Cloud can involve moving large data sets and/or applications across the network infrastructure. With the aim of minimizing the total processing time, the optimal placement of image data and processing algorithms on a large scale, distributed Cloud infrastructure is a challenging task. This work presents a genetic algorithm-based approach for data and application (virtual machine) placement using hypervisor and network metrics to avoid service level agreement violations. The solution involves placing medical image data and associated processing algorithms at optimized processing and compute nodes located within the Cloud. The results of initial experiments show that a genetic algorithm-based placement approach can increase Cloud-based application performance.**

Keywords-cloud computing; virtual machine placement; genetic algorithm; network awareness.

I. INTRODUCTION

The rapid growth in the use of Electronic Health Records (EHR) across the globe along with the rich mix of multi-media held within an EHR combined with the increasing level of detail due to advances in diagnostic medical imaging means increasing amounts of data can be stored for each patient [1][2]. In a scenario where a consultant may view and process medical images remotely for the purpose of producing a diagnosis it may be necessary to move large data sets across the network for processing to take place [3]. Moving such data sets has the potential to introduce undesirable latency and also degrade application performance to an unacceptable level, causing service level agreement (SLA) violations and degrading network performance for other users of the same infrastructure.

Cloud Computing has come to the fore as a new model of computing service delivery as a utility over the Internet. Virtualisation technology [4] lying at the heart of the Cloud allows greater utilisation of physical and virtual resources. Depending on the resources available physical hosts or nodes on the Cloud can host numerous virtual machines, which in turn can host applications and data. Migrating medical imaging applications and data to the Cloud can allow healthcare organisations to realise significant cost savings relating to hardware, software, buildings, power and staff, in addition to greater scalability, higher performance and resilience [5][6]. Cloud Computing uses a 'pay as you go' pricing model whereby users only pay for the amount of resources they consume, e.g., storage, memory, CPU, bandwidth. Additional resources can also be provisioned in an on-demand fashion to allow scaling with application and user demand.

This paper proposes a method for service providers to optimise the combined placement of image processing algorithms (as Virtual Machines - VMs) and image data sets on compute and storage nodes respectively. The state of physical node resources and the network health are given key consideration as critical factors when making placement decisions. The solution uses a genetic algorithm as an initial solution to ensure VMs are placed on nodes, which satisfies SLA and network performance constraints. The results of initial experiments in Section VI show that a genetic algorithm can find optimised solutions, which offer lower total processing cost (image processing and network costs as a function of time) than a random assignment solution. Future work is aimed at improving the convergence time of the genetic algorithm through the design and implementation of a hybrid evolutionary algorithm.

The rest of this paper is organised as follows; Section II describes work relating to optimised VM placement. Section III details a mathematical model of the problem. Section IV defines the design of the proposed solution. Section V describes the initial experiments with results in Section VI. Section VII contains the conclusions and future work.

II. RELATED WORK

Genetic algorithm-based placement solutions have been shown to provide optimised placement in the Cloud [7][8]. Placement of data in Cloud based storage using a genetic algorithm solution has the benefit of reducing the average data access time [7]; however memory, CPU and network constraints are not taken into account in this work. The research presented in [8] is primarily concerned with minimisation of the total execution time and although it does consider network based constraints, critical node constraints such as CPU, memory and storage are not considered. Resource allocation in the Cloud taking CPU and memory requirements in addition to network bandwidth, reliability and throughput requirements has been investigated [9]; but CPU and bandwidth resources are considered as static finite resource with the inability to dynamically scale with demand as and when required. The research outlined above is concerned with the placement of either applications or data independently of one another. Although physical node and

 ISBN: 978-1-61208-216-5

network constraints are taken into account, the placement of application (VM) and associated data is not considered.

Combined application (VM) and data placement taking CPU, memory, storage and network constraints into account has been investigated [10] and a solution using a penalty-based genetic algorithm described; however, the algorithm execution time does show an increase as the number of servers increases, causing a significant delay, which could render it unacceptable if used in a real time solution and may also lead to scalability problems. Hybrid evolutionary algorithms combining the best features of genetic algorithms with the best features of other evolutionary algorithms such as particle swarm optimisation (PSO) [11], ant colony optimisation (ACO) [12], and simulated annealing (SA) [13], have been shown to have a much shorter convergence time than purely genetic algorithm-based solutions [14]. Hybrid genetic algorithms such as the multi agent genetic algorithm [15] can offer superior performance over traditional genetic algorithms when very large scale and dynamic optimisation problems are concerned. Likewise, an improved genetic algorithm (IGA) [16] has been shown to be nearly twice as fast at finding optimised solutions as a purely genetic algorithm placement solution.

III. PROBLEM SPECIFICATION

A. Model Attributes

Processing nodes A and storage nodes B are separated by a network containing a set of network routes R between any set of nodes. A set of virtual machines V containing algorithms are hosted on a set of physical processing nodes A. A set of virtual machines W containing data stores are hosted on a set of physical storage nodes B.

TABLE I. MODEL ATTRIBUTES

Notation	Description
x_{tva}	The placement of task t on vm v on processing node a
y_{dwb}	The placement of dataset d on vm w on storage node b
T	Set of tasks
D	Set of datasets
V	Set of processing virtual machines
W	Set of datastore virtual machines
A	Set of processing nodes
B	Set of storage nodes
R	Set of network routes between nodes a and b
C_a	CPU capacity of processing node a
M_a	Memory capacity of processing node a
S_b	Storage capacity of storage node b
Ct	CPU requirement of task t
M_t	Memory requirement of task t
S_d	Storage requirement of dataset d
C_{tdab}	The cost of task t processing dataset d on nodes a and b
K_a	The network cost between nodes a and b
bw_{ab}	The minimum end to end bandwidth (kbps) of the network path between nodes a and b
lat_{ab}	The network latency (ms)between nodes a and b
T_{sla}	Required response time specified in an SLA

A set of tasks T are executed on a set of processing nodes A. Each processing node a has a resource capacity in terms of CPU C_a and memory M_a. Each task t has resource requirements in terms of CPU C_t and memory M_t. A set of datasets D are stored on a set of storage nodes B. Each storage node b has a resource capacity in terms of storage S_b. Each dataset d has a storage requirement S_d.

B. Mathematical Model

1) Image Processing

$$x_{tva} = \begin{cases} 1 \text{ if task } t \text{ is executed on vm } v \text{ on node } a \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

$$\text{for } t = 1,..,T; v = 1,...,V; a = 1,...,A$$

2) Data Storage

$$y_{dwb} = \begin{cases} 1 \text{ if dataset } d \text{ is stored on vm } w \text{ on node } b \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

$$\text{for } d = 1,...,D; w = 1,...,W; b = 1,...,B$$

3) Objective Function

The aim is to minimise the cost of executing task t on dataset d on processing node a and storage node b – taking the network cost (as a function of time) between a and b into account. Therefore the objective function is to minimise:

$$\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{v=1}^{V}\sum_{w=1}^{W}\sum_{a=1}^{A}\sum_{b=1}^{B} x_{tva}\, y_{dwb}\, c_{tdab} + \sum_{a=1}^{A} k_a \sum_{t=1}^{T}\sum_{v=1}^{V} x_{tva} \quad (3)$$

4) Network Cost

The network cost K_a between processing node a and storage node b is derived from the dataset size S_d divided by the minimum network bandwidth bw_{ab} plus the network latency lat_{ab} on the end to end network route r between node a and node b.

$$k_a = \sum_{r_{a,r,b} \in R} \frac{S_d}{bw_{ab}} + lat_{ab} \quad (4)$$

C. Physical Constraints

1) Processing Constraint - VM to Processing Node

Each task t is executed on a VM v on a processing node a. Each task t has a CPU requirement C_t. Node a must have sufficient CPU capacity C_a to meet the CPU requirement C_t of task t, subject to:

$$\sum_{t=1}^{T}\sum_{v=1}^{V} c_t\, x_{tva} \le c_a \text{ for } a = 1,..., A \quad (5)$$

2) Memory Constraint – VM to Processing Node

Each task t has a memory requirement M_t. Processing node a must have sufficient memory capacity M_a to meet the memory requirement M_t of task t, subject to:

$$\sum_{t=1}^{T}\sum_{v=1}^{V} m_t\, x_{tva} \le m_a \text{ for } a = 1,..., A \quad (6)$$

3) Data storage constraint – VM to Storage Node

Each dataset d has a storage requirement S_d. Each storage node b must have sufficient storage capacity S_b to meet the storage requirement S_d of dataset d, subject to:

$$\sum_{d=1}^{D}\sum_{w=1}^{W} s_d \; y_{dwb} \le s_b \text{ for } b = 1,\ldots,B \quad (7)$$

4) SLA Time Constraint – Data to User

The total processing time must be less than the required response time specified in the SLA T_{sla}, subject to:

$$\sum_{t=1}^{T}\sum_{d=1}^{D}\sum_{v=1}^{V}\sum_{w=1}^{W}\sum_{a=1}^{A}\sum_{b=1}^{B} x_{tva} \; y_{dwb} < T_{sla} \quad (8)$$

D. Logical Constraints

Each task t has one dataset d, subject to:

$$\sum_{d=1}^{D}\sum_{v=1}^{V}\sum_{a=1}^{A}\sum_{w=1}^{W}\sum_{b=1}^{B} x_{tva} \; y_{dwb} = 1 \text{ for } t = 1,\ldots,T \quad (9)$$

Each dataset d has at least one task t, subject to:

$$\sum_{t=1}^{T}\sum_{v=1}^{V}\sum_{a=1}^{A}\sum_{w=1}^{W}\sum_{b=1}^{B} x_{tva} \; y_{dwb} > 1 \text{ for } d = 1,\ldots,D \quad (10)$$

Each VM v is allocated to at least one processing node a, subject to:

$$\sum_{a=1}^{A} x_{tva} \ge 1 \text{ for } v = 1,\ldots,V \quad (11)$$

Each VM w is allocated to at least one storage node b, subject to:

$$\sum_{b=1}^{B} y_{dwb} \ge 1 \text{ for } w = 1,\ldots,W \quad (12)$$

Each task t is executed on at least one VM v on at least one processing node a, subject to:

$$\sum_{v=1}^{V}\sum_{a=1}^{A} x_{tva} \ge 1 \text{ for } t = 1,\ldots,T \quad (13)$$

Each dataset d is stored on at least one VM w on at least one storage node b.

$$\sum_{w=1}^{W}\sum_{b=1}^{B} y_{dwb} \ge 1 \text{ for } d = 1,\ldots,D \quad (14)$$

IV. SOLUTION DESIGN

The aim of the proposed solution is to optimally place data and image processing algorithms on the service provider infrastructure whilst avoiding customer SLA violation. Figure 1 gives an overview of the proposed system. When placing the image processing application CPU, memory, and network constraints need to be satisfied, likewise when placing data a certain amount of storage, adequate network bandwidth and an acceptable latency is required.

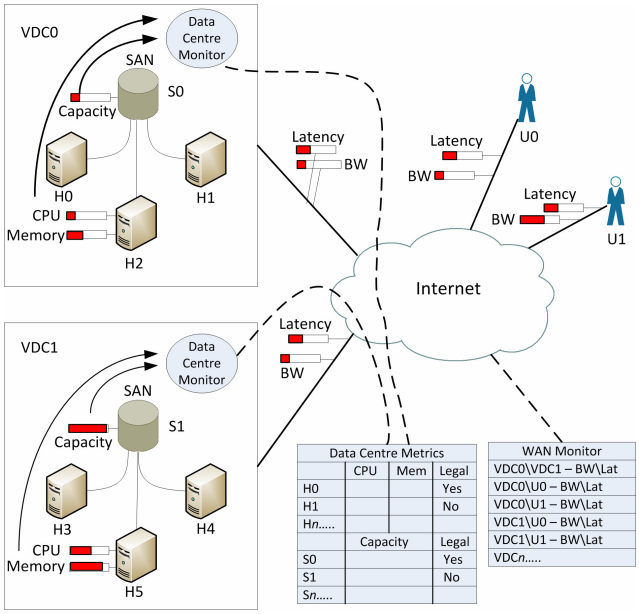

Figure 1. Architectural overview of the proposed system

The 'Data Centre Monitor' is responsible for monitoring the CPU and memory utilisation of hosts (e.g., H0, H1, H2) and the storage capacity of storage area network (SAN) nodes (e.g., S0, S1) within each Virtual Data Centre (VDC). Data centre node metrics are gathered by distributed agents along with network health metrics collected by the 'WAN Monitor', which uses a modified version of BWPing [17] to monitor the end to end bandwidth and latency between all VDCs and users. The node and network health metrics are normalised and form a combined fitness score for each node, which can satisfy the physical and logical constraints. A genetic algorithm is used to find an optimised solution within the pool of viable nodes.

V. INITIAL EXPERIMENTS

A genetic algorithm was developed using Microsoft Visual Studio 2008. A synthetic dataset containing values representing realistic CPU, memory, storage and network metrics for 20 physical nodes was generated. A randomly generated initial population of 50 was used with binary tournament parent selection with a 10% population mutation rate chance. The number of physical nodes was constant at 20, whilst the number of VMs requiring placement increased in increments of 5, ranging from 5 to 75.

Two scenarios were investigated in initial experiments: random placement and genetic algorithm placement. The experiments for each scenario were repeated 30 times and the mean taken. The experiments were conducted on a PC running Windows XP with a 2933 MHz Intel Processor and 4GB of RAM.

 ISBN: 978-1-61208-216-5

VI. RESULTS

The initial results in Figure 2 below show that a genetic algorithm solution (depicted as the lower solid line) produces placement decisions with a lower total processing cost than a random placement solution as depicted by the upper dashed line (initial fittest) in the graph. The costs for each solution are similar when the number of VMs requiring placement are small. Both solutions show a linear increase in cost as the number of VMs requiring placement increases, but the total image processing cost for the genetic algorithm is significantly lower than that of the random placement solution. With a maximum number of 75 VMs for placement the cost associated with random placement is 3229, whilst the genetic algorithm solution is 1294, which is just over 40% of the cost of the random placement solution.

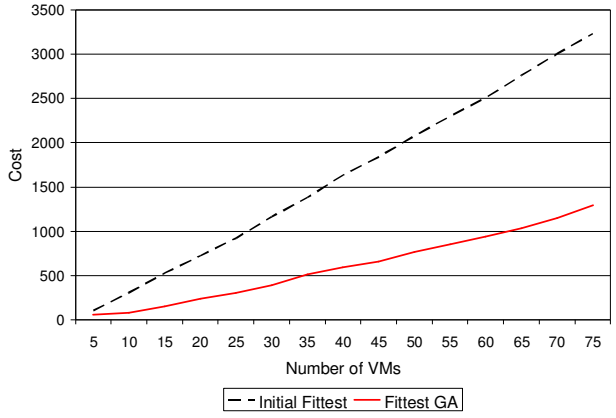

Figure 2. Performance comparison between initial fittest (random placement) and genetic algorithm placement solutions.

VII. CONCLUSION AND FUTURE WORK

A model of VM and data placement including physical node and network constraints was presented. Results from initial experiments show that a genetic algorithm taking multiple constraints into account can be used to make optimised network aware and SLA compliant combined VM and data placement decisions. The total image processing cost was reduced by nearly 60% when compared to a naive random placement solution.

A solution based purely on a genetic algorithm may suffer from scalability issues stemming from long convergence times found in large solution search spaces [10][14], potentially causing unacceptable latency in live systems. Future work will consist of expanding the model to include additional constraints relating to intellectual property (IP) rights. Initial experiments will be scaled to investigate the upper bounds of performance with greater numbers of nodes and VMs, which will be used as an evaluation baseline for future solutions. The development of a hybrid evolutionary algorithm, combining the best features of several evolutionary algorithms will be investigated with the aim of improving performance and resource utilisation.

A prototype system is under development using the NETCOM Cloud testbed facility at the University of Ulster. It will be used to validate current and future results on a dynamic real time Cloud infrastructure.

ACKNOWLEDGMENT

BT EPSRC CASE Award in collaboration with BT EBTIC.

REFERENCES

[1] T. Chia-Chi, J. Mitchell, C. Walker, A. Swan, C. Davila, D. Howard and T. Needham, "A medical image archive solution in the cloud," Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on, pp. 431-434, 16-18 July 2010

[2] H. QingZang, Y. Lei, Y. MingYuan, W. FuLi and L. RongHua, "Medical Information Integration Based Cloud Computing,"Network Computing and Information Security (NCIS), 2011 International Conference on, vol.1, pp. 79-83, 14-15 May 2011

[3] M.I.B. Nordin and M.I. Hassan, "Cloud resource broker in the optimization of medical image retrieval system: A proposed goalbased request in medical application," National Postgraduate Conference (NPC), 2011, pp. 1-5, Sept. 2011

[4] M. Mahjoub, A. Mdhaffar, R.B. Halima and M. Jmaiel, "A Comparative Study of the Current Cloud Computing Technologies and Offers," Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on, pp. 131-134, 21-23 Nov. 2011

[5] L.A. Bastiao Silva, C. Costa, A. Silva and J.L. Oliveira, "A PACS Gateway to the Cloud," Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on, pp. 1-6, 15-18 June 2011

[6] S. Ahmed and A. Abdullah, "E-healthcare and data management services in a cloud," High Capacity Optical Networks and Enabling Technologies (HONET), 2011, pp. 248-252, 19-21 Dec.2011

[7] K. Jindarak and P. Uthayopas, "Performance improvement of cloud storage using a genetic algorithm based placement," Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on, pp. 54-57, 11-13 May 2011

[8] A.V. Dastjerdi, S.K. Garg and R. Buyya, "QoS-aware Deployment of Network of Virtual Appliances Across Multiple Clouds," Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, pp. 415-423, Nov. 29 2011-Dec. 1 2011

[9] K.H. Prasad, T.A. Faruquie, L.V. Subramaniam, M. Mohania and G. Venkatachaliah, "Resource Allocation and SLA Determination for Large Data Processing Services over Cloud," Services Computing(SCC), 2010 IEEE International Conference on, pp. 522-529, 5-10 July 2010

[10] Z.I.M. Yusoh and T. Maolin, "A penalty-based genetic algorithm for the composite SaaS placement problem in the Cloud," Evolutionary Computation (CEC), 2010 IEEE Congress on, pp. 1-8, 18-23 July 2010

[11] J. Kennedy and R. Eberhart. "Particle swarm optimization," In IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995

[12] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey" in Theoretical Computer Science 344 (2–3) (2005), pp.243–278, 2005

[13] S. Kirkpatrick, C.D. Gellatt and M.R. Vechi, "Optimization by simulated annealing". Science, 1983, 220: pp. 671-680

[14] C.C.T. Mark, D. Niyato and C.K. Tham, "Evolutionary Optimal Virtual Machine Placement and Demand Forecaster for Cloud Computing," Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on, pp. 348-355, March 2011

 ISBN: 978-1-61208-216-5

[15] Z. Kai, S. Huaguang, L. Lijing, G. Jinzhu and C. Guojian, "Hybrid Genetic Algorithm for Cloud Computing Applications," Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific, pp. 182-187, 12-15 Dec. 2011

[16] Z. Hai, T. Kun and Z. Xuejie, "An Approach to Optimized Resource Scheduling Algorithm for Open-Source Cloud Systems," ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual, pp. 124-129, 16-18 July 2010

[17] BWPing, http://bwping.sourceforge.net/, accessed 14.05.2012

 ISBN: 978-1-61208-216-5