An Architecture for Measuring Network Performance in Multi-Core Multi-Cluster Architecture (MCMCA)

Norhazlina Hamid, Robert J. Walters, and Gary B. Wills

***Abstract*— The aim of this research is to design a new architecture for large-scale clusters to improve the communication within the interconnection network to gain higher performance. The new architecture will be based on clusters built using workstations containing multi-cored processors in a multi-cluster architecture in the presence of uniform traffic. Multi-core technology is proposed to achieve higher performance without driving up power consumption and heat, which is the main concern in a single-core processor. The architecture will avoid congestion and deadlocks in the network to guarantee faster message transmission. The architecture performance will be validated through simulation, experimental and measurements under various working conditions.**

***Index Terms*—HPC, multi-core cluster, multi-cluster, interconnection network, performance.**

I. Introduction

A computing cluster is typically built from a group of workstations connected by high-speed networks to form a single high-availability system. Overall performance of computing cluster always depends on the efficiency of its communication networks. Hence, performance analysis of the interconnection networks is vital. A general problem in the network may arise from the fact that multiple messages can be in transmission at the same time using the same network links.

Moore's Law, which states that the number of transistors on a processor will double approximately every two years has been proven to be consistent due to the transistors getting smaller in successive processor technologies [1]. However, by increasing the speed with a smaller transistor causes transistors to consume more power and generate more heat [2]. Therefore, computer engineers have designed the multi-core processor, a single processor with two or more cores [3]. This allows the processor to perform more work within a given clock cycle and at the same time reduce unnecessary power consumption [4]. From the combination of these technologies, the multi core cluster architecture has emerged. The multi-core cluster architecture becomes more powerful due to the combination of faster processors, faster memory and faster interconnection [5].

Manuscript received January 30, 2014. The authors acknowledge the award of a Malaysia Fellowship Training scholarship (HLP) to Norhazlina Hamid to allow this research to be undertaken.

N. H. Author is with the School of Electronics & Computer Science, University of Southampton, SO17 1BJ, United Kingdom (e-mail: nh3g11@ecs.soton.ac.uk).

R. J. W. Author is with the School of Electronics & Computer Science, University of Southampton, SO17 1BJ, United Kingdom (e-mail: rjw1@ecs.soton.ac.uk).

G. B. W. Author is with the School of Electronics & Computer Science, University of Southampton, SO17 1BJ, United Kingdom (e-mail: gbw@ecs.soton.ac.uk).

Many studies [6], [7], [8] have been carried out to improve the performance of multi-core cluster but few clearly distinguish the key issue of the performance of interconnection networks. Therefore, the existing models are unable to capture the potential performance of the interconnection networks within an implementation of a multi-core cluster architecture. The cluster interconnection network is critical for delivering efficiency and scalability of the applications, as it needs to handle the networking requirements of each processor core [9].

In a multi-core cluster architecture, multiple computing nodes are connected via the cluster interconnection network. The implementation of the architecture typically imposes higher latency for communication between processors located on different nodes compared with the processors located on the same nodes. A high latency interconnect can dramatically reduce the efficiency of the cluster system [10].

Scalability is always an important aspect to examine when evaluating clusters. Abdelgadir, Pathan, & Ahmed [11] find that having a good network bandwidth and faster network will produce better performance in relation to scalability of clusters. Scaling up by adding more processors per node to increase the speed will create too much heat [3]. The conventional approach to improving cluster throughput is to add more processors but there is a limit to the scalability of this approach; the infrastructure cannot provide effective memory access to unlimited numbers of processors and the interconnection network(s) become saturated [12]. Technological advances have made it viable to overcome these problems by combining multiple clusters of heterogeneous networked resources into what is known as a multi-cluster architecture [13]. This work will expand the architecture to include a scalable approach by applying a multi-cluster architecture. This research is the first investigation into employing multi-core clusters within a multi-cluster architecture.

The rest of the paper is organized as follows: Section 2 briefly introduce multi-core cluster, Section 3 presents the new architecture of the cluster, Section 4 describes the communication network involved in the new architecture, Section 5 describes the research methodologies used, Section 6 records our findings and Section 7 concludes the paper and future work.

II. Background

In the past, several parallel machines with different architectures have been built as viable platforms for High-Performance Computing (HPC) [14], such as distributed shared memory and clusters of multiprocessors. However, with the emergence of high-speed networks, the HPC community has adopted network based computing clusters as cost-effective platforms [15] to achieve high performance.

High performance is a computational activity requiring more than a single computer to execute a task [14]. The trend has been shifting towards cluster systems with multi-core [16], which will be the focus of this paper. The Top500 super-computer list published in November [17] showed that multi-core processors have been widely deployed in clusters of parallel computing, and more than 95% of the systems are using dual-core to quad-core processors. Another motivation in this realm is the advances in multi-core processor technology that makes them an excellent choice to use in cluster architecture [18].

Multi-core means to integrate two or more complete computational cores within a single chip [3]. The motivation of the development of multi-core processors is the fact that scaling up processor speed results in dramatic rise in power consumption and heat generation. In addition, it becomes so difficult to increase processor speed that even a little increase in performance will be costly [7]. Realizing this factor, computer engineers have designed multi-core processors that speed up application performance by dividing the workload among multiple processing cores instead of using one "super-fast" single processor. Due to its greater computing power and cost-to-performance effectiveness, multi-core processor has been deployed in cluster computing [19].

A multi-core cluster is a cluster where all the nodes in the cluster have multi-core processors, as shown in Fig. 1. Each node has multiple processors, each of which contains multiple cores. With such cluster nodes, both the memory and the connection to the interconnection are now shared.

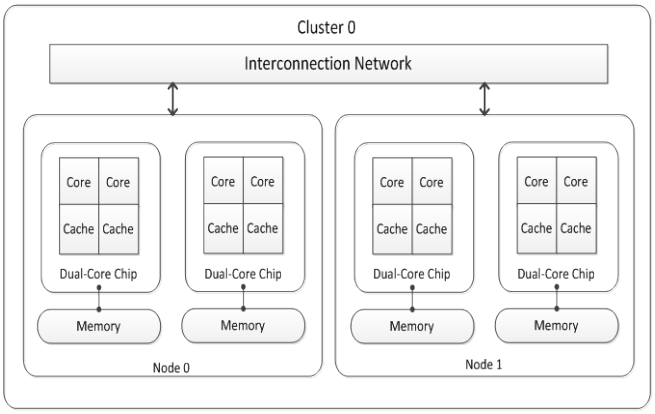

Fig. 1. Illustration of Multi-core Cluster

Multi-core clusters typically have a hierarchical memory structure, where cores from the same processor share L2 or L3 cache [20]. On the other hand, cores belonging to distinct processors within the same node share the main memory (RAM or DRAM) and cores belonging to different nodes do not share any memory resource. High performance can be achieved when executing parallel applications with tasks being allocated to the cores according to the application communication pattern and environment characteristics [21]. Tasks that communicate more frequently should be allocated to the same node avoiding remote communication. However, depending on the amount of task computation and data to be processed, the allocation of multiple tasks to the same processor can be a bottleneck due to the resources being shared by the processor cores [18].

III. The Architecture

A new architecture known as the Multi-core Multi-cluster Architecture (MCMCA) is introduced in Fig. 2. The structure of MCMCA is derived from a Multi-Stage Clustering System (MSCS) [12] which is based on a basic cluster using single core nodes. The MCMCA is built up of numbers of clusters where each cluster is composed of numbers of nodes. The numbers of nodes are determined at run-time. Each node of a multi-core cluster has a number of processors, each with two or more cores with their own L2 cache. Cores on the same chip share the local memory. The interconnection network connects the cluster nodes.

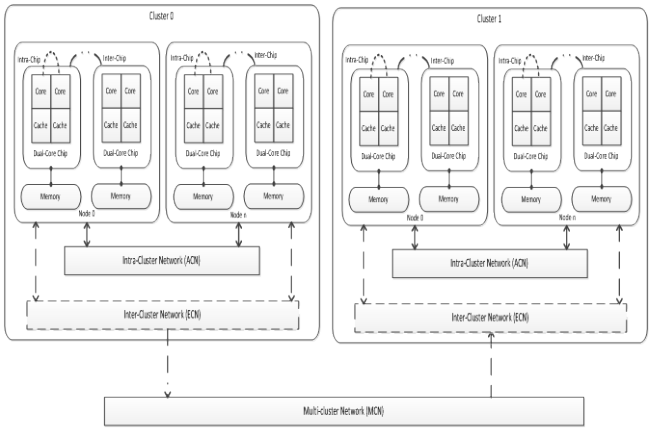

Fig. 2. Overview of the proposed Multi-core Multi-cluster architecture (MCMCA)

IV. Communication Networks

The research conjecture is that low communication latency is essential to achieving a faster network and increasing the efficiency of a cluster. In order to understand the communication network of the Multi-Core Multi-Cluster Architecture (MCMCA), this section explains in detail the different types of communication networks.

This research will focus on five communication networks. There are three communication networks commonly found in a multi-core cluster architecture, these are: the intra-chip communication network (AC); the inter-chip communication network (EC) and the intra-cluster network (ACN). The new communication network introduced in this paper is the inter-cluster network (ECN) and the multi-cluster network (MCN).

The communication between two processor cores on the same chip is the intra-chip communication network (AC), as shown in Fig. 3. Messages from source A to destination B travel via the AC communication network, which acts as a connector between two processor cores on the same chip.

Fig. 4 shows an inter-chip communication network (EC) for communicating across processors in different chips but within a node. Messages travelling to different chips from source A in the same node first have to communicate within the chip via the AC network, and then travel between the chips via the EC network to reach their destination B. Each node has two communication connections which are intra cluster net-

work (ACN) for transmission within a cluster and inter cluster network (ECN) for transmission between clusters.

An intra-cluster network (ACN) is used for messages passing between processors on different nodes but within the same cluster. In order for messages to cross the nodes, messages have to communicate with the AC network and the EC network to pass between chips. Then messages travel via the intra-cluster network (ACN) to enter different nodes to reach their destination, as shown in Fig. 5.

Messages travelling from source A to destination B between clusters communicate via two communication networks to reach other clusters, as shown in Fig. 6. An inter-cluster network (ECN) is used to transmit messages between clusters, as well as for the management of the entire system. The clusters connect to each other via the multi-cluster network (MCN). When the messages reach the other cluster, they will have to communicate with the ECN of the target cluster before arriving at their destination.

All levels of communication are critical in order to optimise the overall performance of the multi-core multi-cluster architecture (MCMCA). The overall communication latency gathered from all communication networks will be calculated. The derived simulation results will be analysed for comparison between the existing architecture and the MCMCA architecture.

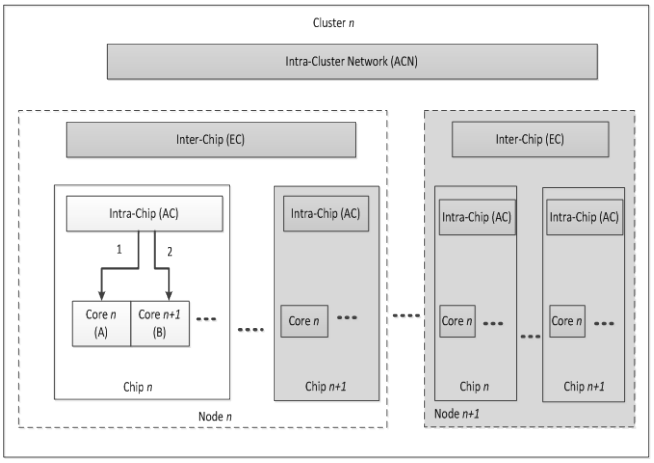

Fig. 3. Communication network flow A for message passing between two processor cores on the same chip

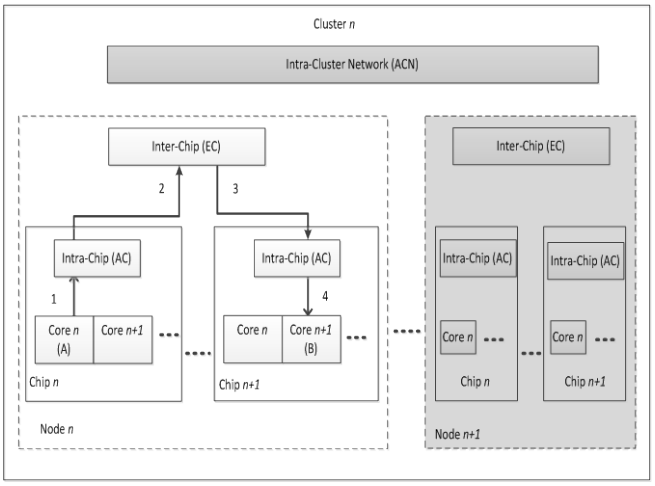

Fig. 4. Communication network flow B for message passing across processors in different chips but within a node

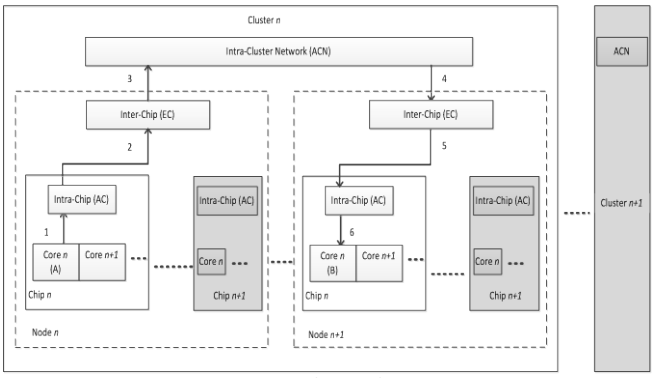

Fig. 5. Communication network flow C for message passing between processors on different nodes but within the same cluster

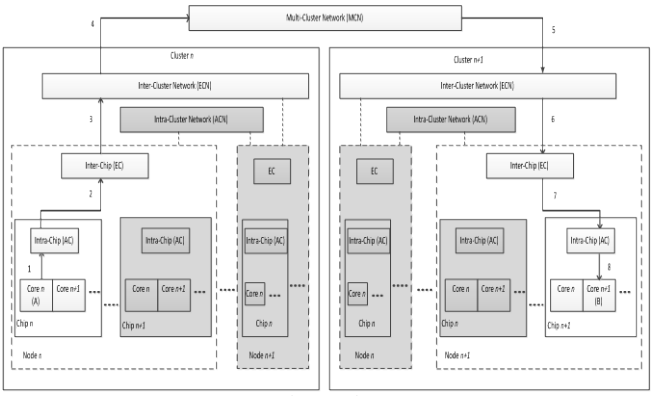

Fig. 6. Communication network flow D for transmitting messages between clusters

V. RESEARCH METHODOLOGY

This work will concentrate on computer simulation to model the architecture. The accuracy of the model will be validated through computer simulation experiments. OMNeT++ network simulation tool has been chosen to model the multi-core clusters in the multi-cluster architecture. OMNeT++ [22] is a C++-based open-source discrete-event simulator which uses the process-interaction approach and has been publicly available since 1997. OMNeT++ can be used for the simulation of computer networks and distributed or parallel systems.

An early stage of simulation experiments under various configurations and design parameters has been completed. The performance evaluation focused on communication latency in the MCMCA architecture. As a preliminary study, the communication network performance and experiment are based on a single-core multi-cluster architecture. A simulation model has been developed to measure the performance of single-core multi-cluster architecture. The evaluation was then compared to the model of multi-cluster architecture presented by Javadi, Akbari, & Abawajy [23] with the given configuration and parameters to match the work in their papers. The configuration of the simulation was based on the list of interconnection network parameter in Table 1.

TABLE I: INTERCONNECTION NETWORK PARAMETER [23]

Parameter	*Intra-cluster (ACN)*	*Inter-cluster (ECN)*
Network Latency	0.01s	0.02s
Switch Latency	0.01s	0.01s
Network Bandwidth	1000b/s	500b/s

This work focuses on measuring steady-state performance of a network; the performance of a network with a stationary traffic source after it has reached steadiness. A network has reached steadiness when its average queue lengths have reached their steady-state values. To measure steady-state performance, the simulation experiments were conducted in three phases: warm-up, measurement and drain [9]. The network has necessarily reached a steady-state once the network is warmed up [24]. This means that the statistics of the network are stationary and no longer changing with time, which will determine an accurate estimation. Statistics were gathered in each simulation experiment.

The simulation model is built on the basis of the following assumptions which are used in similar studies [23], [24], [25]:

1) The underlying system is a large-scale cluster with two types of communication networks: intra-cluster network and inter-cluster networks.
2) Each processor generates packets independently, following a Poisson distribution with a mean rate of lambda (λ) and inter-arrival times are exponentially distributed.
3) The destination of each message is any node in the system with uniform distribution.
4) The numbers of processors and cores in all clusters are the same and the cluster nodes are homogeneous.
5) The communication switches are input buffered and each channel is associated with a single packet buffer.
6) Message length is fixed.

VI. Results and Discussion

The simulation of the single-core multi-cluster simulation model has been examined with a number of cases. When the simulation is started, a message will travel in the network following the routing algorithm which will determine the path from the source to the destination. A long message will be divided into one or more packets and each packet will be partitioned into a sequence of flits, a flow control digits, to make sure the resources can be allocated directly to the messages.

The performance under various workload conditions has been evaluated with the first case was performed for an 8-single-core cluster system with message length (M) = 32 flits, flit length (F) = 256 bytes and 512 bytes. The second case was performed with the same 8-single-core cluster system and the same flit length (F) = 256 bytes and 512 bytes but with longer message length (M) = 64 flits.

Results in Fig. 7 are derived from computer simulation based on the first case with message length (M) = 32 flits while Fig. 8 depicts the results of the second case with message length (M) = 64 flits. The X axis of the graph represents the traffic generation rate, while the Y axis denotes the communication latency.

Simulation experiments have revealed that the results obtained from the single-core multi-cluster architecture closely match the results from the model of multi-cluster architecture presented by Javadi et al. [23], when compared. The results have shown that as the traffic rate increases, the average communication latency increases following the assumptions that the messages have to wait for resources before traversing into a network. At low traffic rates, latency will approach zero-load latency. The zero-load latency assumption is that a packet never contends for network resources with other packets. The results confirm that the simulation model is a good basis to measure the communication latency for a large-scale cluster, and can be extended to multi-core multi-cluster architecture.

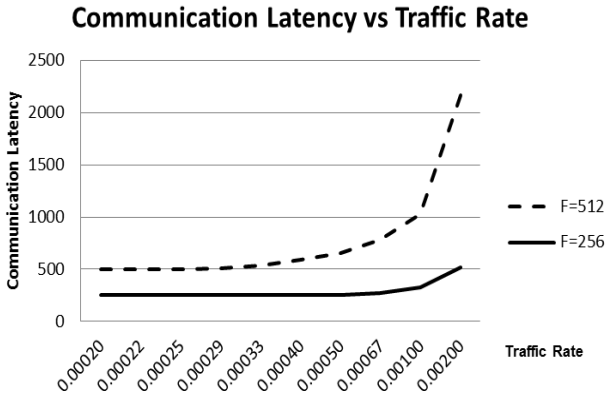

Fig. 7. Average latency of simulation model for single-core multi-cluster architecture of 8-cluster system with M=32 flits, F=256 bytes and 512 bytes

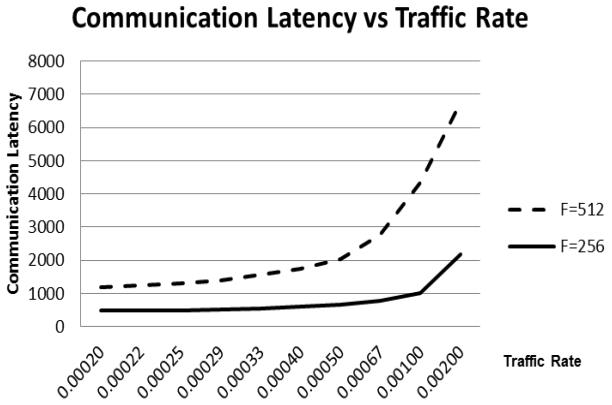

Fig. 8. Average latency of simulation model for single-core multi-cluster architecture of 8-cluster system with M=64 flits, F=256 bytes and 512 bytes

VII. Conclusion

In conclusion, this paper presented new architecture for measuring interconnection network performance in multi-core cluster based on multi-cluster architecture. This architecture will predict an average communication latency of a cluster when applying multi-core multi-cluster architecture. The validity of the architecture will be demonstrated by computer simulation and experimental measurements within the simulation.

Our future work will be developing a simulation model for MCMCA. The simulation will measure communication latency of a cluster when applying multi-core processor technology, under a multi-cluster architecture environment. The performance measurements will focus on overall communication latency within the simulation model and the simulation results will be analysed for comparison with published results of existing cluster architectures [23], [26].

Acknowledgment

We thank Dr. Bahman Javadi for his guidance in simulation development.

REFERENCES

[1] Intel. (1997, Moore's Law and Intel Innovation. Available: http://www.intel.com/about/companyinfo/museum/exhibits/moore.htm?wapkw=moore+laws

[2] D. Geer, "For Programmers, Multicore Chips Mean Multiple Challenges," *Computer*, vol. 40, pp. 17-19, 2007.

[3] T. W. Burger, "Intel Multi-Core Processors: Quick Reference Guide," 2005.

[4] N. Karmakar, "Multi-core Architecture," North Maharashrta University, India2011.

[5] E. W. Bethel and M. Howison, "Multi-core and many-core shared-memory parallel raycasting volume rendering optimization and tuning," *International Journal of High Performance Computing Applications*, vol. 26, pp. 399-412, November 1, 2012 2012.

[6] S. Ichikawa and S. Takagi, "Estimating the Optimal Configuration of a Multi-Core Cluster: A Preliminary Study," in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on*, 2009, pp. 1245-1251.

[7] C. Lei, A. Hartono, and D. K. Panda, "Designing High Performance and Scalable MPI Intra-node Communication Support for Clusters," in *Cluster Computing, 2006 IEEE International Conference on*, 2006, pp. 1-10.

[8] A. Ranadive, M. Kesavan, A. Gavrilovska, and K. Schwan, "Performance implications of virtualizing multicore cluster machines," presented at the Proceedings of the 2nd workshop on System-level virtualization for high performance computing, Glasgow, Scotland, 2008.

[9] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Network*: Morgan Kaufmann, 2004.

[10] G. Shainer, P. Lui, M. Hilgeman, J. Layton, C. Stevens, W. Stemple, *et al.*, "Maximizing Application Performance in a Multi-core, NUMA-Aware Compute Cluster by Multi-level Tuning," in *Supercomputing*. vol. 7905, J. Kunkel, T. Ludwig, and H. Meuer, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 226-238.

[11] A. T. Abdelgadir, A.-S. K. Pathan, and M. Ahmed, "On the Performance of MPI-OpenMP on a 12 nodes Multi-core Cluster," in *Algorithms and Architectures for Parallel Processing*, ed: Springer, 2011, pp. 225-234.

[12] H. S. Shahhoseini, M. Naderi, and R. Buyya, "Shared memory multistage clustering structure, an efficient structure for massively parallel processing systems," in *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, 2000, pp. 22-27 vol.1.

[13] J. H. Abawajy and S. P. Dandamudi, "Parallel job scheduling on multicluster computing system," in *Cluster Computing, 2003. Proceedings. 2003 IEEE International Conference on*, 2003, pp. 11-18.

[14] Y. Qian, "Design and Evaluation of Effiecient Collective Communications on Modern Interconnects and Multi-core Clusters," 2010.

[15] L. Hope and E. Lam, "A Review of Applications of Cluster Computing," *World*, pp. 1-10, n.d.

[16] X. Wu and V. Taylor, "Performance modeling of hybrid MPI/OpenMP scientific applications on large-scale multicore supercomputers," *Journal of Computer and System Sciences*, 2013.

[17] Admin. (2009, TOP500 Highlights - November 2009. Available: http://www.top500.org/lists/2009/11/highlights

[18] M. Soryani, M. Analoui, and G. Zarrinchian, "Improving inter-node communications in multi-core clusters using a contention-free process mapping algorithm," *The Journal of Supercomputing*, pp. 1-26, 2013/04/10 2013.

[19] L. Chai, "High Performance and Scalable MPI Intra-node Communication Middleware for Multi-core Clusters," PhD, Graduate School of The Ohio State University, The Ohio State University, 2009.

[20] S. Fengguang, S. Moore, and J. Dongarra, "Analytical modeling and optimization for affinity based thread scheduling on multicore systems," in *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on*, 2009, pp. 1-10.

[21] J. M. N. Silva, L. Drummond, and C. Boeres, "On Modelling Multicore Clusters," in *Computer Architecture and High Performance Computing Workshops (SBAC-PADW), 2010 22nd International Symposium on*, 2010, pp. 25-30.

[22] A. Varga. (2001). *OMNeT++*.

[23] B. Javadi, M. K. Akbari, and J. H. Abawajy, "A performance model for analysis of heterogeneous multi-cluster systems," *Parallel Computing*, vol. 32, pp. 831-851, 2006.

[24] B. Javadi, J. H. Abawajy, and M. K. Akbari, "A comprehensive analytical model of interconnection networks in large-scale cluster systems," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 75-97, 2008.

[25] W. Yulei, M. Geyong, L. Keqiu, and B. Javadi, "Modeling and Analysis of Communication Networks in Multicluster Systems under Spatio-Temporal Bursty Traffic," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, pp. 902-912, 2012.

[26] C. Lei, G. Qi, and D. K. Panda, "Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System," in *Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on*, 2007, pp. 471-478.

Norhazlina Hamid received a Bachelor in Information Technology (Hons) from Northern University of Malaysia (UUM) in 2000 and an MSc in Information Technology from MARA University of Technology (UiTM) in 2003. She is now a final year PhD student in the School of Electronics and Computer Science of University of Southampton.

Dr Robert Walters worked for almost fifteen years working in commercial banking, before leaving to study Mathematics with Computer Science at University of Southampton. After completing his degree he worked for several years as a software developer before returning to Southampton as a Research Fellow in 1996. Since then he has completed his PhD in 2003 and is currently employed as a Lecturer in the School of Electronics and Computer Science of University of Southampton. His research interests include middleware, distributed computing, hypermedia and graphical formal modelling language.

Dr Gary Wills is a Senior Lecturer in Computer Science at the University of Southampton. He graduated from the University of Southampton with an Honours degree in electromechanical engineering, and then a PhD in Industrial hypermedia systems. He is a Chartered Engineer and a member of the Institute of Engineering Technology and a Fellow of the Higher Educational Academy. He is also a visiting professor at the Cape Peninsular University of Technology, SA. Dr. Gary's main research interests are in Personal Information Environments (PIEs) and their application to industry, medicine and education. PIE systems are underpinned by Service Oriented Architectures, adaptive systems and advanced knowledge technologies.