

---

# An Analytical Model of Multi-Core Multi-Cluster Architecture (MCMCA)

Norhazlina Hamid, Robert John Walters, Gary Brian Wills

School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK  
{nh3g11, rjw1, gbw}@ecs.soton.ac.uk

---

## ABSTRACT

*Multi-core clusters have emerged as an important contribution in computing technology for provisioning additional processing power in high performance computing and communications. Multi-core architectures are proposed for their capability to provide higher performance without increasing heat and power usage, which is the main concern in a single-core processor. This paper introduces analytical models of a new architecture for large-scale multi-core clusters to improve the communication performance within the interconnection network. The new architecture will be based on a multi - cluster architecture containing clusters of multi-core processors.*

## TYPE OF PAPER AND KEYWORDS

Short communication: *multi-core processor, multi-core cluster, analytical analysis, performance model, interconnection networks*

## 1 INTRODUCTION

The emergence of High Performance Computing (HPC), which includes cloud computing and cluster computing, has improved the availability of powerful computers and high speed network technologies. It can be concluded that the main target of HPC is better performance in computing. HPC aims to leverage cluster computing to solve advanced computation problems. While cluster computing has been widely used for scientific tasks, cloud computing was originally intended to serve business applications. Dillon et al. [1] have pointed out that the current cloud is not geared for HPC for several reasons. Firstly, it has not yet matured enough for HPC; secondly, unlike cluster computing, cloud infrastructure only focuses on enhancing the system performance as a whole; thirdly, HPC aims to enhance the performance of a specific scientific application using resources across multiple organisations. The key difference from cloud

computing is in elasticity: for cluster computing the capacity is often fixed, while running an HPC application can often require considerable human interaction, e.g. tuning based on a particular cluster with a fixed number of homogenous computing nodes [2]. This is contrasted with the self-service nature of cloud computing, in which it is hard to know how many physical processors are needed. In order to achieve higher availability and scalability of applications executed within cloud resources, it is important to supplement the capabilities of management services with high performance cluster computing to enable full control over communication resources.

Cloud computing has changed the way both software and hardware are purchased and used. An increasing number of applications is becoming web-based since such applications are available from anywhere and from any device. These applications are using the infrastructures of large-scale data centres and

can be provisioned efficiently. Hardware, on the other side, representing basic computing resources, can also be delivered to match the specific demands without the user/consumer having to actually own them. As more organisations adopt clouds, the need of high availability platforms and infrastructures, the cluster, to facilitate and distribute the load across multiples processor is evolving [3] [4].

The Top 500 supercomputer list published in Jun 2014 [5] showed that multi-core processors have been widely deployed in clusters of parallel computing, and more than 96% of the systems are using six or more core processors. Several performance models have been proposed in literature to improve the performance of multi-core clusters but few clearly distinguish the key issue of the communication performance of interconnection networks [6] [7] [8] [9]. Therefore, the existing models are unable to capture the potential communication 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 [10]. The novelty allows organizations to develop a cluster-based private cloud to improve efficiency and reduce job submission failure [11].

Multi-core means to integrate two or more complete computational cores within a single chip [12]. The motivation of the development of multi-core processors is from the fact that scaling up processor speed results in a 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, the multi-core processor has been deployed in cluster computing [13].

Many studies [6] [7] [8] have been carried out to improve the performance of multi-core clusters but few clearly distinguish the key issue of the performance of interconnection networks. Although the cluster interconnection network is critical for delivering efficient performance, as it needs to handle the networking requirements of each processor core [14], existing models do not address the potential performance issues of the interconnection networks within multi-core clusters.

Abdelgadir, Pathan and Ahmed [15] find that having a good network bandwidth and a faster network will produce a better performance in relation to the

scalability of the clusters. 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 networks become saturated [16]. 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 introduces multi-core multi-cluster architecture, Section 3 presents the analytical model of the architecture, Section 4 presents the analytical implementation, Section 5 describes the results and findings and Section 6 summarizes and concludes the paper.

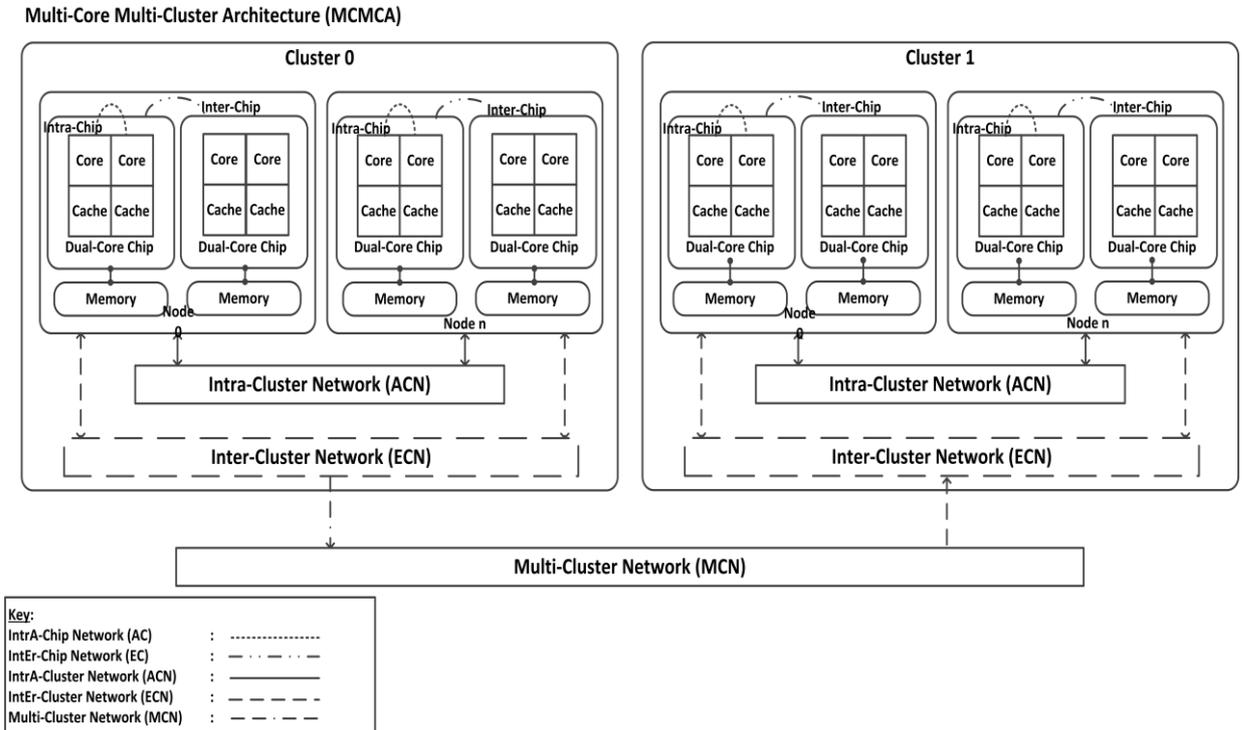
## **2 MULTI-CORE MULTI-CLUSTER ARCHITECTURE (MCMCA)**

A multi-core cluster is a cluster, where all the nodes in the cluster have multi-core processors. In addition, each node may have multiple processors (each of which contains multiple cores). With such cluster nodes, the processors in a node share both memory and their connections to the outside.

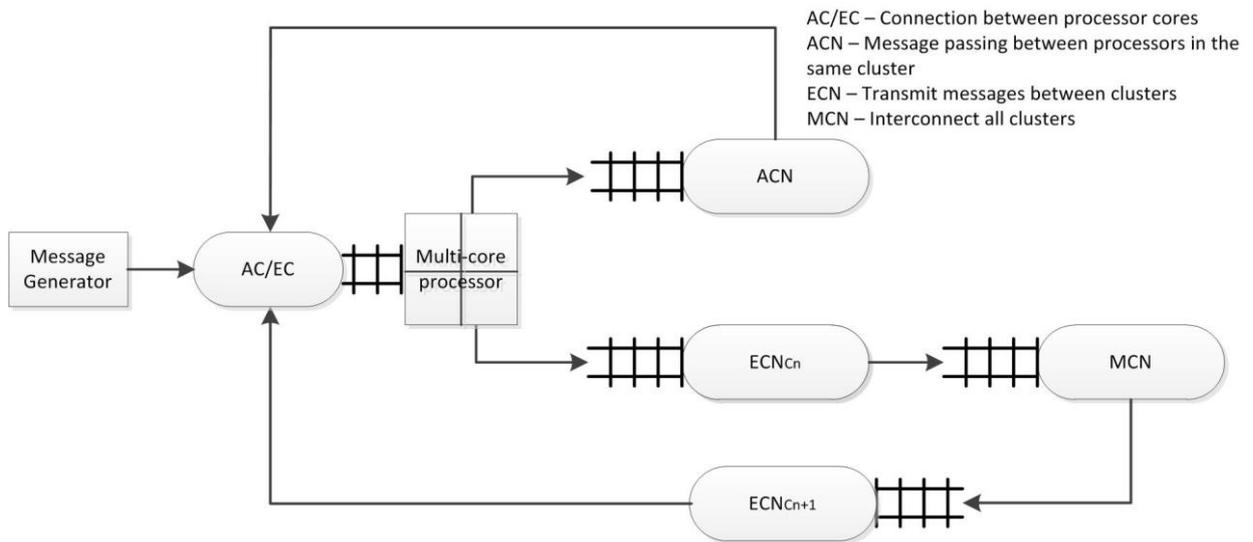
A new architecture known as the Multi-Core Multi-Cluster Architecture (MCMCA) is introduced in Figure 1. The structure of MCMCA is derived from a Multi-Stage Clustering System (MSCS) [16], which is based on a basic cluster using single-core nodes. The MCMCA is built up of a number of clusters, where each cluster is composed of a number of nodes. Each node of a cluster has a number of processors, each with two or more cores. Cores on the same chip share the local memory and the cluster nodes are connected through the interconnection network.

### **2.1 Queuing Network Model**

Message passing in Multi-Core Multi-Cluster Architecture (MCMCA) is embedded with the queuing network model approach as shown in Figure 2. Approximations of packet latency are based on queuing model to predict the average amount of time that a packet spends waiting in each queue in the architecture. A queuing network consists of service centers (i.e., processor cores) and customers (i.e., packets). A service center has one or more queues to hold jobs waiting for service. After being serviced, a job either moves to another service center or exits the network.



**Figure 1. Overview of the proposed Multi-Core Multi-Cluster Architecture (MCMCA)**



**Figure 2. MCMCA's Queuing Model**

In MCMCA interconnection networks, packets spend a lot of time waiting in queues before they are allowed to travel to their destination. A source will generate packets with a rate of  $\frac{1}{\lambda}$  packets per second. The packets will stay in a queue while waiting to be transmitted by a processor core. A processor core then removes the packets from the queue on a first-in-first-out (FIFO) basis and processes them with an average transmission time.

This paper will consider the distribution of the transmission time upon reaching a high traffic due to a packet's arrival in an M/G/1 queuing network. M/G/1 queuing networks are used to analyze systems with Poisson arrival and exponentially distributed transmission time [17].

## 2.2 Routing Algorithm and Switching Method

The routing algorithm and switching method are important components of an interconnection network. The routing algorithm establishes the path between the source and the destination of a message. The proposed model will adopt a deterministic routing algorithm applied by Bahman's model based on the well-known Up\*/Down\* routing [18], where a message traveling from the source node to the destination node will go up through internal switches of the tree until it finds the Nearest Common Ancestor (NCA) and then is transmitted down to the destination node. In this algorithm, each message experiences two phases, an ascending phase to get a nearest common ancestor (NCA), followed by a descending phase. The deterministic routing algorithm balances the traffic distribution and will extinguish the switch contention problem [19]. In the deterministic routing, a message traverses a fixed path between the source and the destination, which simplifies the implementation, avoids message deadlock and guarantees in-order delivery [20].

The switching method determines the way that packets travel from switch to switch in other paths or levels. The store-and-forward switching has risen in popularity in cluster systems due to its ability to achieve optimal performance in terms of the throughput [21]. In the store-and-forward switching, a message is divided into a sequence of packets and each packet is sent along a path such that the entire message is received by each switch on the path (store) before it is sent to the next switch on the path (forward). The store-and-forward switching allows the utilisation of the full bandwidth for every connection and can quickly release connections as soon as messages have

passed the connection, and this reduces the risk of deadlocks [22].

## 2.3 Interconnection Networks

An interconnection network is a connection between two or more computer networks via network devices such as routers and switches, to exchange traffic back and forth and guide traffic across the complete network to their destination [23]. Routers will determine the route for a packet based on a routing algorithm and transmit it from the source to its destination of a node on another network. When a packet has to travel from one interconnection network to another to get to its destination, many problems can arise. The method each interconnection network uses to cross the network may be different from one to another, and this may contribute to communication latency of interconnection network.

The performance of the architecture depends on the communication latency of its interconnection networks. The research conjecture is that a low communication latency is essential to achieving a faster network and increasing the efficiency of a cluster. There are five communication networks in Multi-Core Multi-Cluster Architecture (MCMCA) [24] [25]. Three of them are commonly found in any multi-core cluster architecture, and these are: the intra-chip communication network (AC); the inter-chip communication network (EC) and the intra-cluster network (ACN). The new communication networks introduced in this paper are the inter-cluster network (ECN) and the multi-cluster network (MCN).

### 2.3.1 Intra-Chip network (AC)

The communication between two processor cores on the same chip is the intra-chip network (AC), as shown in Figure 3. Messages will be divided into numbers of cores by the AC network, which acts as a connector between two or more processor cores on the same chip. Dividing the messages into a number of cores, in theory, result in more than twice the performance with lower communication delay [26].

### 2.3.2 Inter-Chip network (EC)

Figure 4 shows an inter-chip network (EC) for communicating across processors in different chips but still within the same node. Messages travelling to different chips in the same node will communicate via the intra-chip (AC) and inter-chip (EC) to reach their destination.

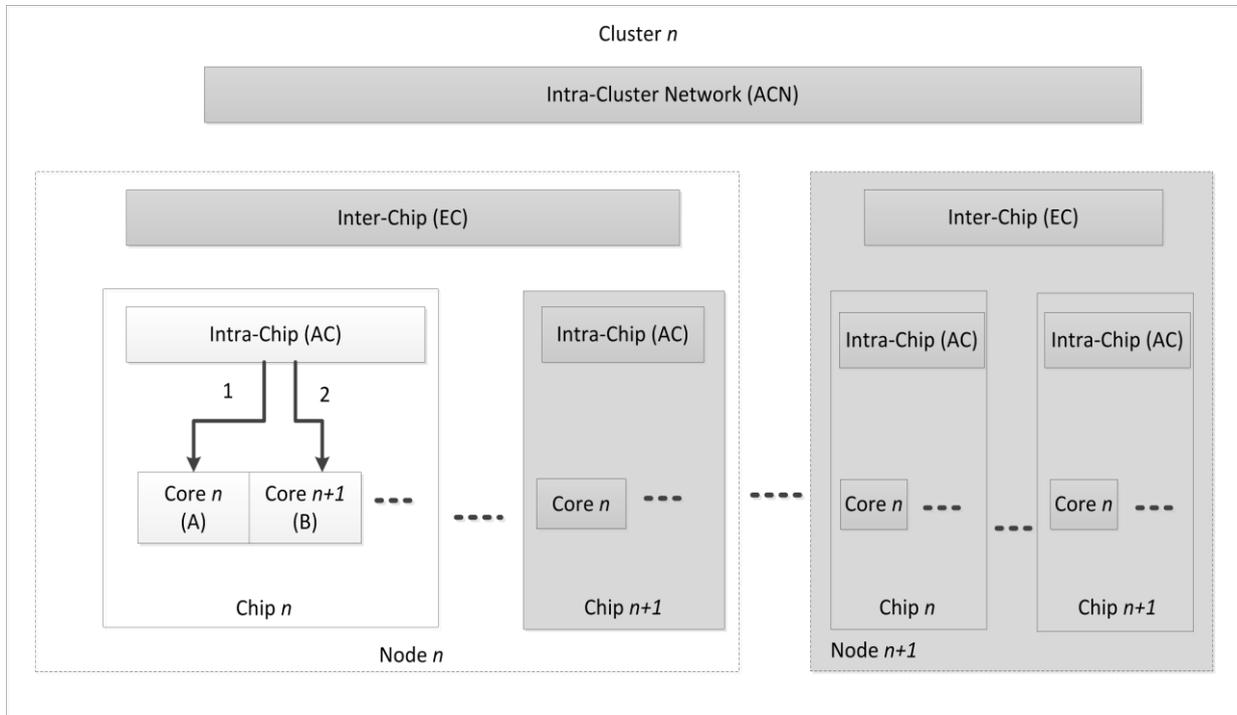


Figure 3. Communication for message passing between two processor cores on the same chip

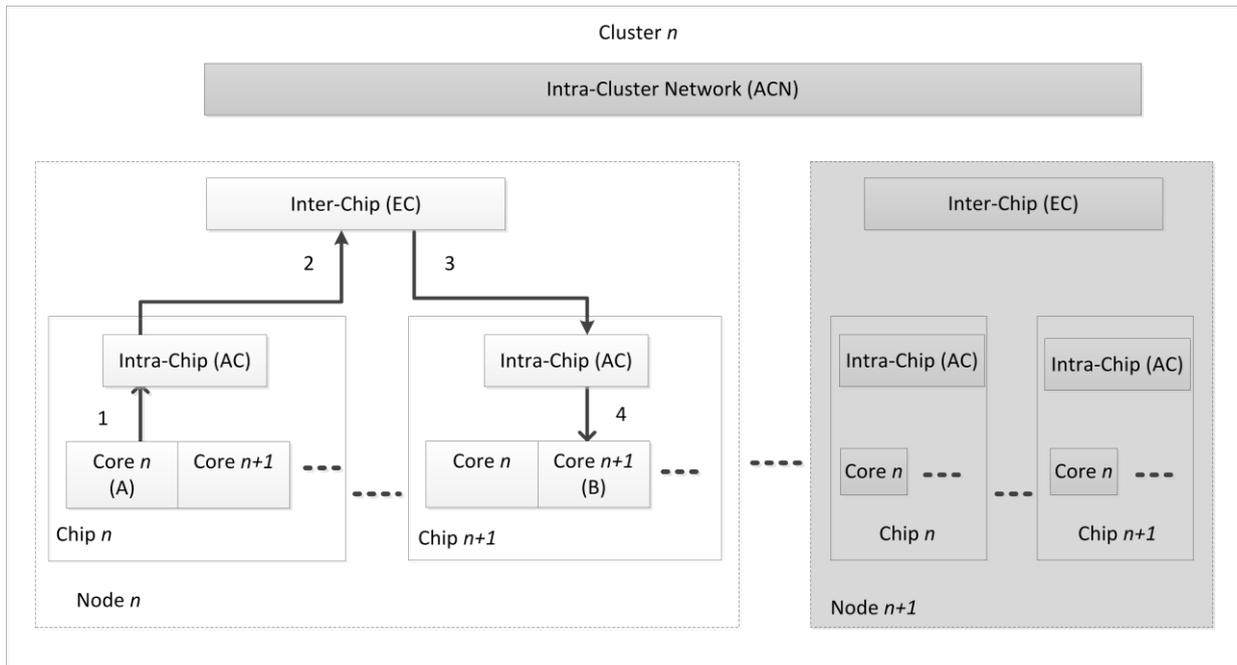


Figure 4. Message passing across processors in different chips, but within a node

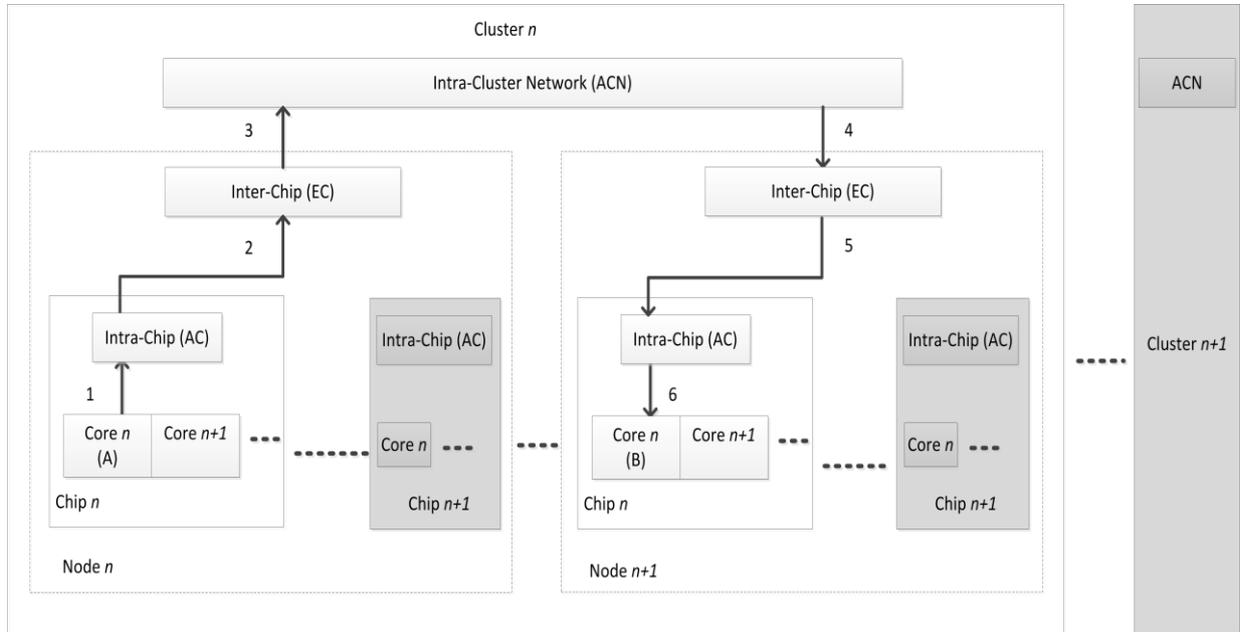


Figure 5. Communication for message passing between processors on different nodes, but within the same cluster

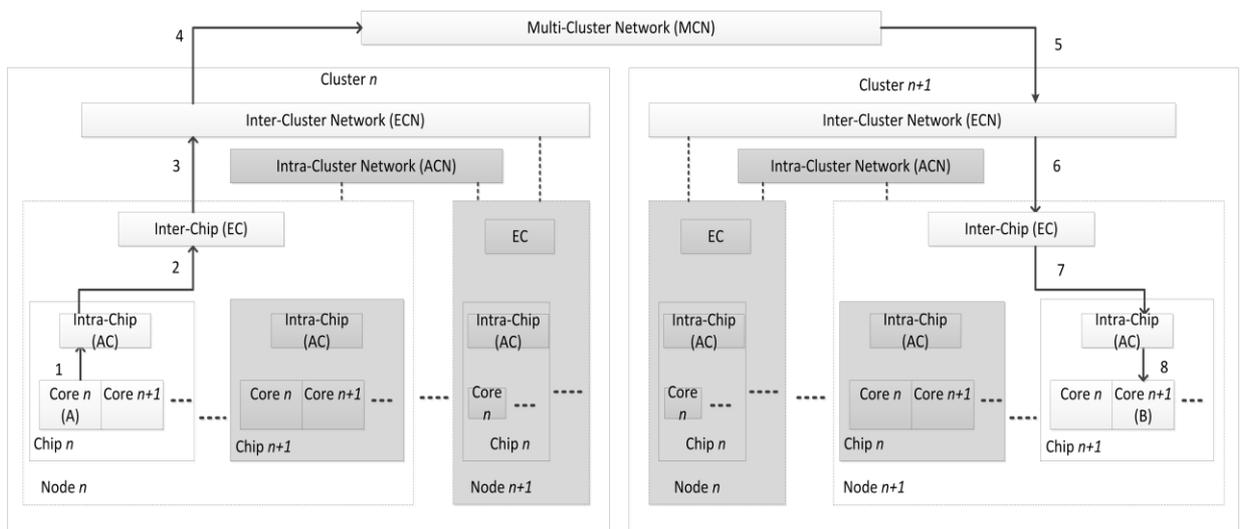


Figure 6. Communication for transmitting messages between clusters

### 2.3.3 Intra-Cluster Network (ACN)

Intra-cluster network (ACN) is an interconnection network to connect nodes within a cluster. Messages that cross the nodes to other nodes in the same cluster will be connected by ACN via intra-chip (AC) and the inter-chip (EC) to complete its journey, as shown in Figure 5.

### 2.3.4 Inter-Cluster Network (ECN) and Multi-Cluster Network (MCN)

The longest route for messages to travel will involve ECN and MCN. Messages travelling from their source to their destination between clusters communicate via two interconnection networks to reach other clusters, as shown in Figure 6. An inter-cluster network (ECN) is used to transmit messages between clusters. The clusters are connected to each other via the multi-cluster network (MCN). When the messages reach the other cluster, it will be connected by the ECN of the target cluster before arriving at its destination. The same process will continue to the other clusters until all the packets exit the network.

## 3 THE ANALYTICAL MODEL

The analytical model is a set of equations describing the performance of a computer system. Analytical models are constructs used to gain an understanding of the current activity on the system, to measure performance and analyse the behaviour of the workloads and hardware within it [27].

Communication networks in MCMCA are divided into internal-cluster and external-cluster, and communication networks latency in the architecture will be determined by four factors:

1. Average waiting time at the source node
2. Average transmission delay for a message to cross the networks
3. Average time for the last packet of the message to reach its destination
4. Average waiting time at transfer switch (external-cluster only)

### 3.1 Assumptions

The model is built on the basis of the following assumptions, which have been used in similar studies [20, 28]:

1. Each processor generates packets independently, following a Poisson distribution with a mean rate of  $\lambda$  and inter-arrival times are exponentially distributed.
2. The destination of each message is any node in the system with uniform distribution.
3. The number of processors and cores in all clusters are the same and the cluster nodes are homogeneous.
4. The communication switches are input-buffered and each channel is associated with a single packet buffer.
5. Message length is fixed.

### 3.2 Average Waiting Time at the Source Node (WT)

Messages injected from a source node enter an internal-cluster network with the probability  $(1 - P)$ . Thus, the traffic arriving at a source node channel is modelled as an M/G/1 queueing model. The waiting time of a message ( $WT_{int}$ ) before entering the network with  $\lambda_{int}$  message arrival rate can be calculated as:

$$WT_{int} = \frac{\lambda I (tsI)^2}{2(1 - \lambda I. tsI)} \quad (1)$$

$$\lambda_{int} = \left(\frac{1}{\lambda}\right) (1 - P) \quad (2)$$

Messages generated by the source nodes are sent to the external-cluster with the probability of outgoing request,  $P$  with  $\lambda_{ext}$  message arrival rate. The waiting time in the external-cluster network ( $WT_{ext}$ ) can be computed by:

$$WT_{ext} = \frac{\lambda E (tsE)^2}{2(1 - \lambda I. tsE)} \quad (3)$$

$$\lambda_{ext} = 2 \left(\frac{1}{\lambda}\right) P \quad (4)$$

$$P = \frac{N - NP}{N - 1} \quad (5)$$

$NP$  is the number of processors in each cluster,  $nc$  is the number of cores in the processors,  $C$  is the number of clusters and  $m$  is the number of ports.

$$NP = 2nc \left(\frac{m}{2}\right)^2 \quad (6)$$

### 3.3 Average Transmission Time for a Message to Cross the Networks (TT)

Each message may use a different number of channel links to reach its destination. Therefore, the transmission time in internal-clusters can be considered as a  $2j$ -channel with  $j$ -channel in the source cluster and  $j$ -channel in the destination cluster through ACN. Similar to internal-clusters, each external message needs to traverse a  $2j$ -channel in ECN and a  $2h$ -channel in MCN to reach its destination. The probability of a message trip to reach its destination,  $P(j, n)$  can be computed by:

$$P(j, n) = \begin{cases} \frac{\left(\frac{m}{2} - 1\right) \left(\frac{m}{2}\right)^{j-1}}{2 \left(\frac{m}{2}\right)^n - 1}, & 1 \leq j < n \\ \frac{(m-1) \left(\frac{m}{2}\right)^{j-1}}{2 \left(\frac{m}{2}\right)^n - 1}, & j = n \end{cases} \quad (7)$$

The number of stages in internal-clusters and external-clusters are determined by  $SSI = 2j - 1$  and  $SSE = 2(j + h) - 1$ . Since this architecture applies store-and-forward flow control, blocking does not happen. Thus, the average transmission time is  $TT = tn$ .

### 3.4 Average Time for the Last Packet of the Message to Reach its Destination (RT)

The equation to calculate the average time for the last packet to reach its destination in the cluster,  $RT$ , is as follows:

$$RT_{int} = \sum_{f=1}^{nc} \sum_{j=1}^n \left[ P_{f, nc} P_{j, n} \left( \sum_{s=1}^{SSI-1} tsI + tnI \right) \right] \quad (8)$$

where,

$$P(f, nc) = \begin{cases} \frac{\left(\frac{m}{2} - 1\right) \left(\frac{m}{2}\right)^{f-1}}{2 \left(\frac{m}{2}\right)^{nc} - 1}, & 1 \leq f < nc \\ \frac{(m-1) \left(\frac{m}{2}\right)^{f-1}}{2 \left(\frac{m}{2}\right)^{nc} - 1}, & f = nc \end{cases} \quad (9)$$

$$RT_{ext} = \sum_{j=1}^n \sum_{h=1}^{nt} \left[ P_{j, n} P_{h, nt} \left( \sum_{s=1}^{SSE-1} tsE + tnE \right) \right] \quad (10)$$

where,

$$P(j, n) = P(h, nt) \quad (11)$$

$tn = 1/2\alpha net + M\beta net$  is the time for a packet of messages to transmit from a node to a switch or vice versa connection while  $ts = \alpha sw + M\beta net$  is the time for a packet of the message to transmit on a switch to switch connection.  $M$  is the message length,  $\alpha net$  and  $\alpha sw$  are the network and switch latency, while  $\beta net$  is the transmission time of one byte and should be calculated as the inverse of the bandwidth.  $nt$  is the number of trees in the MCN.

$$nt = \left\lceil \frac{(\log_2 C) - 1}{(\log_2 m) - 1} \right\rceil \quad (12)$$

### 3.5 Average Waiting Time at Transfer Switches (WTsw)

External-cluster messages need to cross transfer switches during their journeys traversing the network. The transfer switches act as simple buffers to combine traffic from one cluster to other clusters. The waiting time at these buffers,  $WTsw$  with  $\lambda sw$  message arrival rate, can be computed as:

$$WTsw = \frac{\lambda sw (tsE)^2}{2(1 - \lambda sw \cdot tsE)} \quad (13)$$

$$\lambda sw = NP \left( \frac{1}{\lambda} \right) P \quad (14)$$

Therefore, the equations for message latency in the internal-cluster and external-cluster communication networks can be expressed as:

$$Lint = WT_{int} + TT_{int} + RT_{int} \quad (15)$$

$$Lext = WT_{ext} + TT_{ext} + RT_{ext} + 2WTsw \quad (16)$$

From equations (15) and (16), the average message latency of communication networks in the multi-core multi-cluster architecture can be obtained by the sum of the message latency in internal-cluster and external-cluster as follows:

$$TL = Lint (1 - P) + Lext (P) \quad (17)$$

## 4 IMPLEMENTATION OF THE ANALYTICAL MODEL

Algorithm 1 presents the implementation of the analytical model to compute the communication latency of interconnection networks in MCMCA.

<b>Algorithm 1 : Process flow in calculating the communication latency of interconnection networks in MCMCA</b>	
Input Parameter: Number of clusters ( $C$ ), parameter of m-port n-tree, message length ( $M$ ), number of cores ( $nc$ ), number of nodes ( $N$ ) and lambda ( $\frac{1}{\lambda}$ )	
1.	Calculate $P$ , $NP$ and $nt$ using (5), (6) and (12)
2.	Calculate $\lambda_{int}$ , $\lambda_{ext}$ and $\lambda_{sw}$ using (2), (4) and (14)
3.	Calculate $P(j, n)$ and $P(f, nc)$ using (7) and (9)
4.	Calculate $tn = 1/2\alpha_{net} + M\beta_{net}$ and $ts = \alpha_{sw} + M\beta_{net}$ for the internal and external cluster
5.	Calculate average latency in internal-cluster: a. Calculate $WT_{int}$ , the waiting times at the source node based on (1) b. Calculate $RT_{int}$ , the time for the last packet of the message reach its destination using (8)
6.	Calculate average latency in external-cluster: a. Calculate $WT_{ext}$ , the waiting times at the source node based on (3) b. Calculate $RT_{ext}$ using (10) c. Calculate $WT_{sw}$ , the waiting time at the transfer switch using (13)
7.	Calculate the message latency in internal-cluster and external-cluster using (15) and (16)
8.	Calculate $TL$ , the average message latency of interconnection networks in MCMCA using (17)

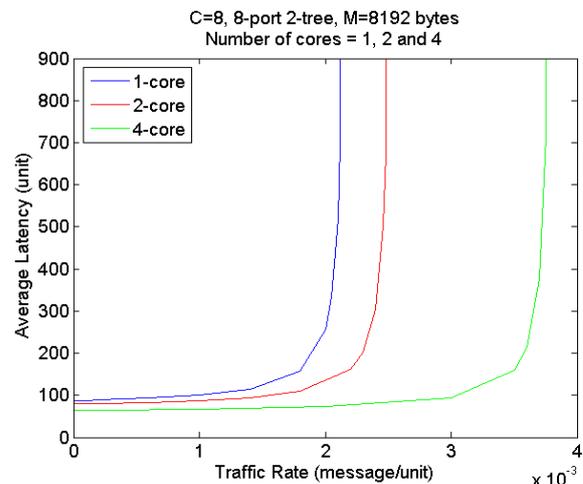
## 5 RESULTS AND FINDINGS

Analysis has been done with three different numbers of cores in a processor. Figure 7 depicts the analytical results when the number of cores equals to 1, 2 and 4. The analysis is investigated using the interconnection network parameter as in Table 1.

**Table 1: Interconnection Network Parameter [29]**

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

The throughput of the network tends to increase as the number of cores is increased. The probability of packet transmits in internal-cluster increased 51%-76% with 2 and 4 cores in each processor compared to single-core processor. This demonstrates that more packets can be transmitted at the same traffic rate, which will save the waiting queue.



**Figure 7. MCMCA for 8-cluster with M=32 with number of cores = 1, 2 and 4**

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 multi-core multi-cluster architecture where the number of cores is equal to 1. A simulation model has been developed to

measure the performance of the MCMCA architecture. The evaluation was then compared to the published model presented by Javadi, Akbari, & Abawajy [29] with the given configuration and parameters to match the work in their papers.

Figure 8 and Figure 9 shows the simulation results of the new architecture for two different sizes of the cluster, 32-cluster with messages length (M) = 32 and 8-cluster with message length (M) = 64 using the same given configuration in Table 1, and the same instances as a Bahman’s model in Table 2. As the traffic increases, the increased contention causes the latency to increase as messages must wait for the buffers and channels, but at a low traffic the latency approaches zero-load latency. The zero-load latency assumption is that a packet has never contended for network resources with other packets. It gives a lower bound on the average latency of a packet through the network. These figures reveal that the latency results obtained from the MCMCA, where the number of cores was equal to 1, closely matched those obtained from Bahman’s model.

Table 2 : Model cases [29]

C, m, n	Message Length (M)	Flit length (F)
32, 8, 2	32 flits	256 bytes
32, 8, 2	32 flits	512 bytes
8, 8, 2	64 flits	256 bytes
8, 8, 2	64 flits	512 bytes

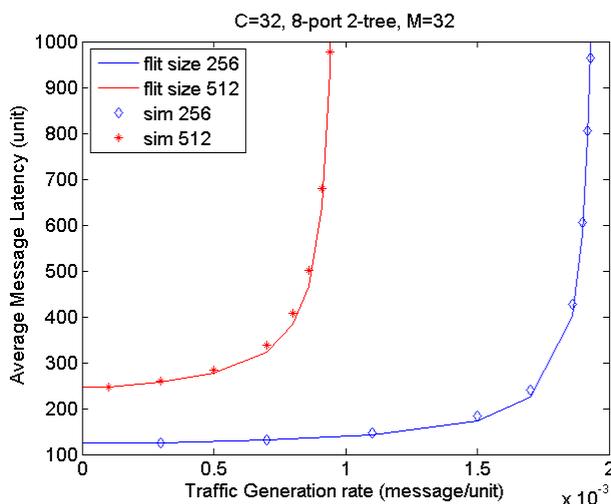


Figure 8. MCMCA for 32-cluster system with M=32 with number of cores = 1

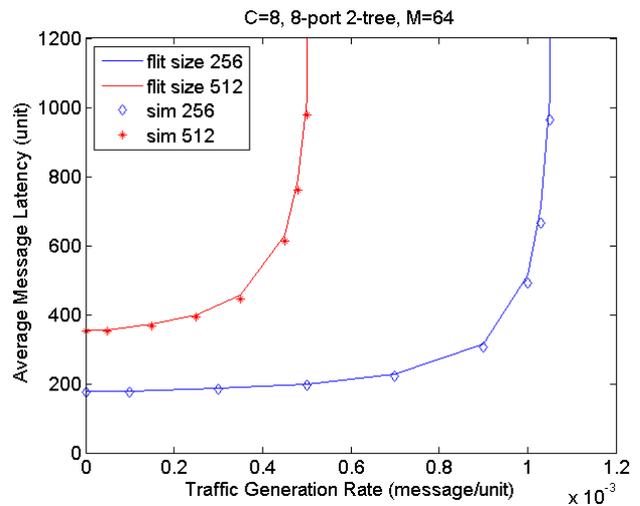


Figure 9. MCMCA for 8-cluster system with M=64 with number of cores = 1

## 6 SUMMARY AND CONCLUSIONS

This paper has presented an analytical model for measuring the performance of interconnection networks in Multi-Core Multi-Cluster Architecture (MCMCA). The analytical model experiments have been conducted with different numbers of cores and baseline results have been produced. The analytical results have shown that the performance of the interconnection network can optimize as the number of cores increase. The results also demonstrated that the architecture can achieve lower communication latency of the interconnection networks at the same traffic rate. The comparison between the analytical results and those produced from the simulation experiments has shown that the derived analytical model possesses a good basis in predicting the communication delay of interconnection network performance of the Multi-Core Multi-Cluster Architecture (MCMCA), which supports the infrastructure as a service for organizations adopting cloud and cluster computing.

## 7 ACKNOWLEDGEMENTS

The authors acknowledge the award of a Malaysia Fellowship Training scholarship (HLP) to Norhazlina Hamid to allow this research to be undertaken.

## 8 REFERENCES

- [1] T. Dillon, W. Chen, and E. Chang, "Cloud Computing: Issues and Challenges", in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pp. 27-33, 2010.
- [2] L. Schubert, K. Jeffery, and B. Neidecker-Lutz, "The Future of Cloud Computing: Opportunities For European Cloud Computing Beyond", Expert Group Report, European Commission, Information Society and Media, 2010.
- [3] V. Chang, R. J. Walters, and G. Wills, "Review of Cloud Computing and existing Frameworks for Cloud adoption," *Advances in Cloud Computing Research*, 2014.
- [4] J. Kosinska, J. Kosinski, and K. Zielinski, "The Concept of Application Clustering in Cloud Computing Environments: The Need for Extending the Capabilities of Virtual Networks", in *Proceedings of the Fifth International Multi-Conference on Computing in the Global Information Technology*, pp. 139-145, 2010.
- [5] Admin. *Top500 Supercomputer Sites*, <http://www.top500.org/lists/2014/06/>, 2014.
- [6] S. Ichikawa and S. Takagi, "Estimating the Optimal Configuration of a Multi-Core Cluster: A Preliminary Study", in *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 1245-1251, 2009.
- [7] C. Lei, A. Hartono, and D. K. Panda, "Designing High Performance and Scalable MPI Intra-node Communication Support for Clusters", in *Proceedings of the IEEE International Conference on Cluster Computing*, pp. 1-10, 2006.
- [8] A. Ranadive, M. Kesavan, A. Gavrilovska, and K. Schwan, "Performance implications of virtualizing multicore cluster machines", in *Proceedings of the 2nd workshop on System-level virtualization for high performance computing*, pp. 1-8, 2008.
- [9] X. Wu and V. Taylor, "Performance modeling of hybrid MPI/OpenMP scientific applications on large-scale multicore supercomputers", *Journal of Computer and System Sciences*, vol. 79, pp. 1256-1268, 2013.
- [10] W. J. Dally and B. P. Towles, "Principles and practices of interconnection networks", *Morgan Kaufmann*, 2004.
- [11] V. Chang, R. Walters, and G. Wills, "Cloud Storage and Bioinformatics in a Private Cloud Deployment: Lessons for Data Intensive Research", in *Cloud Computing and Services Science*. vol. 367, pp. 245-264, 2013.
- [12] T. W. Burger, "Intel Multi-Core Processors: Quick Reference Guide", <https://software.intel.com/en-us/articles/intel-multi-core-processors-quick-reference-guide>, 2005.
- [13] L. Chai, "High Performance and Scalable MPI Intra-node Communication Middleware for Multi-core Clusters," PhD dissertation, Graduate School of The Ohio State University, The Ohio State University, 2009.
- [14] 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, pp. 226-238, 2013.
- [15] 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*, pp. 225-234, 2011.
- [16] H. S. Shahhoseini, M. Naderi, and R. Buyya, "Shared memory multistage clustering structure, an efficient structure for massively parallel processing systems", in *Proceedings of the Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, pp. 22-27, 2000.
- [17] M. Sadeghi and M. Barati, "Performance analysis of Poisson and Exponential distribution queuing model in Local Area Network", in *Proceedings of the International Conference on Computer and Communication Engineering (ICCCCE)*, pp. 499-503, 2012.
- [18] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, *et al.*, "Autonet: a high-speed, self-configuring local area network using point-to-point links", *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1318-1335, 1991.
- [19] 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.
- [20] W. Yulei, M. Geyong, L. Keqiu, and B. Javadi, "Modeling and Analysis of Communication

- Networks in Multicluster Systems under Spatio-Temporal Bursty Traffic", *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 902-912, 2012.
- [21] B. Javadi, J. H. Abawajy, and M. K. Akbari, "Modeling and analysis of heterogeneous loosely-coupled distributed systems", Technical Report TR C06/1, School of Information Technology, Deakin University, Australia, 2006.
- [22] T. Rauber and G. Runger, "Parallel Programming for Multicore and Cluster Systems", *Springer*, 2010.
- [23] A. S. Tanenbaum, "Computer Networks", *Simon & Schuster*, 1996.
- [24] N. Hamid, R. J. Walters, and G. B. Wills, "An architecture for measuring network performance in multi-core multi-cluster architecture (MCMCA)," *International Journal of Computer Theory and Engineering*, vol. 7, pp. 57-61, 2015.
- [25] N. Hamid, R. J. Walters, and G. B. Wills, "Performance evaluation of multi-core multi-cluster architecture (MCMCA)", in *Proceedings of the Emerging Software as a Service and Analytics*, pp. 46-54, 2014.
- [26] H. Furhad, M. A. Haque, C.-H. Kim, and J.-M. Kim, "An Analysis of Reducing Communication Delay in Network-on-Chip Interconnect Architecture", *Wireless Personal Communications*, pp. 1-17, 2013.
- [27] G. V. Caliri, "Introduction to Analytical Modeling", in *Proceedings of the 26th International Computer Measurement Group Conference*, 2000, pp. 31-36, 2000.
- [28] B. Javadi, J. H. Abawajy, and M. K. Akbari, "Performance modeling and analysis of heterogeneous meta-computing systems interconnection networks", *Computers & Electrical Engineering*, vol. 34, pp. 488-502, 2008.
- [29] 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.

## AUTHOR BIOGRAPHIES



**M.Sc. 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.



**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. 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.