A Multi-core architecture for a hybrid information system

Norhazlina Hamid[1], Victor Chang[2], Robert John Walters[3], Gary Brian Wills[3]

1. Malaysian Government, Malaysia

2. IBSS, Xi'an Jiaotong-Liverpool University, Suzhou, China Email: victorchang.research@gmail.com

3. School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK Email: gbw@ecs.soton.ac.uk

Abstract

This paper demonstrates our proposed Multi-core architecture for a hybrid information system (HIS) with the related work, system design, theories, experiments, analysis and discussion presented. Different designs on clusters, communication between different types of chips and clusters and network queuing methods have been described. Our aim is to achieve quality, reliability and resilience and to demonstrate it, our emphasis is on latency with messages communicated in our system – understand how it happens, what can trigger its increase, and then experiment with different types of focuses, including under Store-and-Forward Flow Control method, Wormhole flow control method, cluster size and message size to get a better understanding. Our analysis allows us to reduce latency and avoid its sharp increase. We justify our research contributions, particularly in the area of "traffic analysis and management" and "performance analysis of transmission control" of the HIS systems.

Keywords: Multi-core architecture for a HIS system (MCAHS); Multi-core clusters; latency and message latency for clusters; quality, reliability and resilience (QRR) for HIS systems

1. Introduction

This paper presents hybrid information system (HIS) based on multi-cored clustering system. The objective is to achieve quality, reliability and resilience (QRR). The Multi-core clusters have the advantages of performance improvement and a better co-ordination with hardware and software to ensure a high energy efficiency, a high job completion rate and a low job failure rate. In this way, the services can always function at the optimum level without suffering the quality of service (QoS) [1]. Issues of awareness for QRR include the loss of energy, higher failure rate and loss of data while using a large scale computational powers and resources to maintain a good QoS. In order to achieve this, a hybrid system will require to consolidate from hardware design first and then software design to fix "the root of problems" properly. Hence, the first step is to design an energy-saving and efficient multi-core systems. In our context of grid, cluster and cloud computing, QRR is essential for the success of the service and project as follows [2-3]. First, quality can ensure all jobs can be requested at any time and can be completed at any time. The level of service always stays optimum. Second, reliability means that job

requested can be completed successfully with a high completion rate. All the outputs can be trustworthy to the scientists and stakeholders. Third, resilience can ensure results can always be ready, reproduced and consistent. Different types of tests can produce positive outcomes to support the validity and quality of the service. In this paper, we demonstrate a Multi-core architecture for a HIS system (MCAHS), with the related work, system design, theories, experiments, analysis and discussion presented and explained. Structure of this paper is as follows. Section 2 presents related work including two recommended methods adopted. Section 3 describes different clusters and a proposed Queuing Model. Section 4 illustrates performance evaluation with results and analysis. Section 5 is a Conclusion and Future work to justify our research contributions.

2. Related Work

Clusters consists of different workstations networked together with a domain network, often they are designed to perform specific tasks, such as running scheduled jobs, executing automated tasks and performing analysis. Clusters can be used with cloud computing if virtualized servers with cloud infrastructure can be managed in the data centers. In the old systems, single-core clusters have been used in services that high-speed performance is not the main issue such as storage and backup [4]. There are also hybrid systems that use both multi-core and single-core systems that the demands on performance can be responsible by multi-core systems and the reliability of storage and data safety can be handled by single-core architectures.

While programing execution does not always have the clear advantage on the multi-ore systems, reasons are as follows. First, there are no direct message channels that can be passed directly between the cores, within the chip with multi-cores, between different machines and between different clusters [5]. Second, some programming languages do not specifically design it for multi-threaded or additional work is required. In our approach, messages can be freely passed on between cores, on the chip with multi-cores, between different chips of multi-cores and between different clusters. System design and experiments will be followed with the aim to demonstrate the validity and effectiveness of our approach. In order to understand network theories related to our hybrid systems, related theories are as follows.

2.1 Store-and-Forward Flow Control Method and supporting formulas

Store-and-Forward Flow Control Method is used in packet switching to allow information to be passed on in a unit of four, and then to the next sequence. The cycle can be repeated continuously [6]. This allows network traffic to keep flowing in the best possible ways, since a longer queue may easily result in congestion, as shown in Figure 1. Our hybrid system has adopted this method throughout our architecture. This method also requires

the following criteria to be fulfil. First, each message length is fixed. Second, the cluster nodes are homogeneous with the same number of cores. Third, each message is equally distributed in any node.

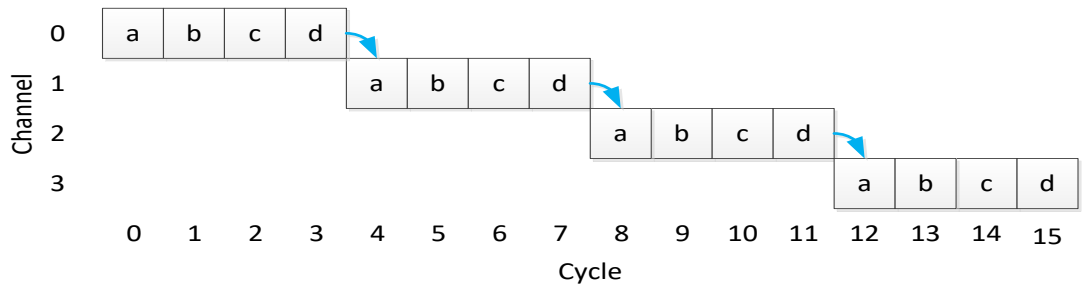

Figure 1: Illustration of Store-and-Forward Flow Control Method

This section presents formulas for network theory related to our proposal. Important variable includes the number of clusters (*C*), number of cores (*nc*), *m*-port *n*-tree and message lengths (m), and ρ is the number of processors in each cluster which can be determined by formula (1)

$$\rho = 2nc\left(\frac{m}{2}\right)^2 \tag{1}$$

Additionally, the packets can be distributed into a designated number of cores throughout the complete cycle, with the same number of cluster nodes based on message probability by given equation. Then P_o, the probability for a message to exit from a cluster, and P_i , the probability of messages staying in a cluster and their relation can be presented by formula (2) and (3).

$$P_o = \frac{N-\rho}{N-1} \tag{2}$$

$$P_i = 1 - P_o \tag{3}$$

$$\alpha_{ie} = 0.5\,\alpha net + M\frac{1}{\beta net} \tag{4}$$

$$\beta_{ie} = \alpha sw + M\frac{1}{\beta net} \tag{5}$$

Execution time to complete tasks are important as follows: α is the time required for information to transmit on a node–to-switch (or vice versa) connection, while β is the time for information to transmit on a switch-to-switch connection. M is the message length, αnet and αsw are the network and switch latencies, and βnet is the transmission time of one byte. They are presented in formula (4) and (5). Experiments and analysis will be described in Section 3.

2.2 Wormhole flow control method

Wormhole flow control has been used extensively in cluster systems due to its low buffering [7]. Wormhole forwards a packet as soon as the header is received, and channel and buffers allocated to flits are acquired without waiting for the entire packet to be received. Thus, packets are divided into a sequence of fixed-size units called 'flits', with channel and buffers allocated to flits. Wormhole flow control makes far more efficient use of buffer space, although it will increase some throughput [7]. In order to improve the efficiency, Chang and Wills [8] have developed a similar method that can process and manage big data processing and improve traffic flow during simulations. Some of key lessons learned in this research can be used as resources to replicate experiments and compare performance with Store-and-Forward Flow Control. Results and analysis will be presented in Section 3.

3. System Design

This section describes system design from the multi-core system. Each node contains the two processors, and each processor contain dual-core chip to process information and computer commands, and the other two remaining sections can accommodate buffering system cache for performance enhancement. A reason is because some job failure is caused by an overloaded system cache and the system is unable to clear them, and if doing so, it can cause the QoS down due to slow performance and the possibility that a system is in a short halt. Each core is then directly connected to the memory so that the system can reduce time to transfer data and data processing to memory. There are also other featured designs as follows. First, the multi-core design supports multi-threading and parallel programming, so that the developers can retrieve and store data directly to memory and CPU-core. Second, the multi-core design can manage cache better and adjust the cache volume for the optimum level of the system. Third, the multi-core design has the improved memory system as it can directly communicate with memory. Last, cooling and energy efficient management system can function at the hardware level, starting from the core.

3.1 Cluster architecture

Figure 2 shows the proposed clustering system. There is a cluster interconnection network and switch built to connect all different clusters together. A cluster contains a group of personal computers (PC)/workstations, operating systems, network interfaces and all the related software, as shown in Figure 2. Multi-core systems are

contained in each workstation. To manage each cluster, either software interfaces or command-line based services can be used. Each cluster interconnected work is specially built to provide the second layer of quality, reliability and resilience (QRR). The speed will need to be optimum with a large bandwidth to ensure there is a low network latency and a low possibility for high network traffic. A robust design similar to Chang [9] can be tested several times for different types of experiments, such as a low network latency, optimum speed at the peak time, low data loss and a drop in QoS through a longer distance. It can provide elasticity and scalability, so if the size of cluster expands, it can be easily adjusted.

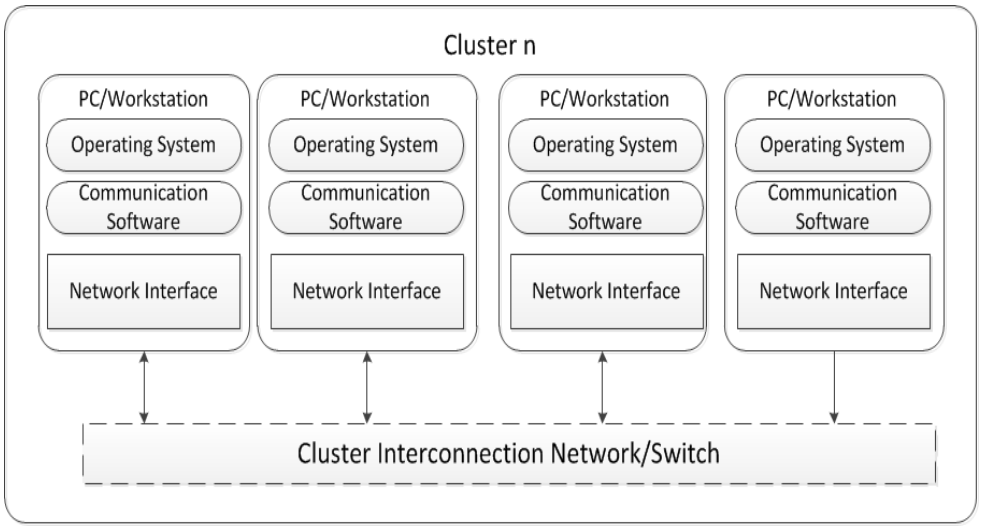

Figure 2: A cluster architecture

To illustrate the concept of a multi-core cluster architecture, Figure 3 shows the example as the basis for our hybrid information system (HIS). The shorter distance and a quicker access between processors, memory, node and cluster network, means the time can be reduced and there is a better data transfer between all these.

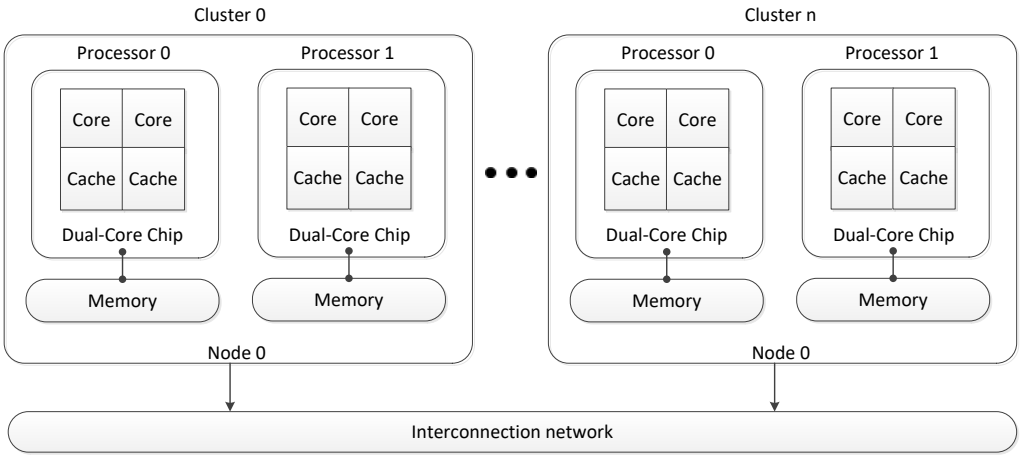

Figure 3: The multi-core architecture as the basis of our HIS system

To ensure there is a better and faster communication within the cluster and between different clusters, our HIS system has been designed in such a way. First, there is an “inter-chip” connection between each core and between each processor. The aim is to ensure programming codes or job requests at the embedded level, can directly communicate between each core and between each processor. Second, there is an “inter-node” between each node, so that communications can be direct without going through more layers or using routing techniques to sort out the shortest communication path. In our architecture, HIS is undertaken starting from the core, node and network in our architecture to optimize the communication and time.

To facilitate all changes, the proposed architecture, Multi-core architecture for a HIS system (MCAHS), has been proposed and developed, as shown in Figure 4. It has the interconnection network in place, so that each processor, each node, each intra cluster network and inter cluster network can freely communicate with each other to reduce message time, improve performance and reduce latency in between them, regardless of being involved in processing commands, data transfer or job requests and completion.

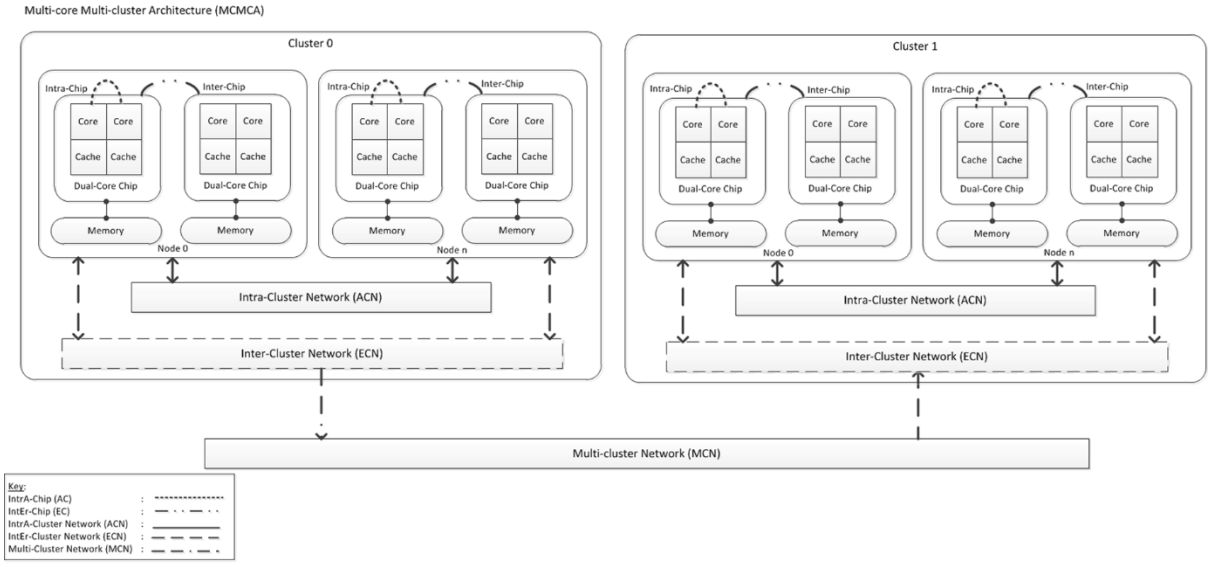

Figure 4: Multi-core architecture for a HIS system (MCAHS)

To demonstrate how intra-chip, inter-chip, intra cluster and inter cluster networks function, Figure 5, 6, 7 and 8 are shown. Figure 5 shows intra-chip network (AC) with the shadow region showing how information can be passed between two processor cores on the same chip. This allows communication in a serial way so that information and data can be passed from one to the next.

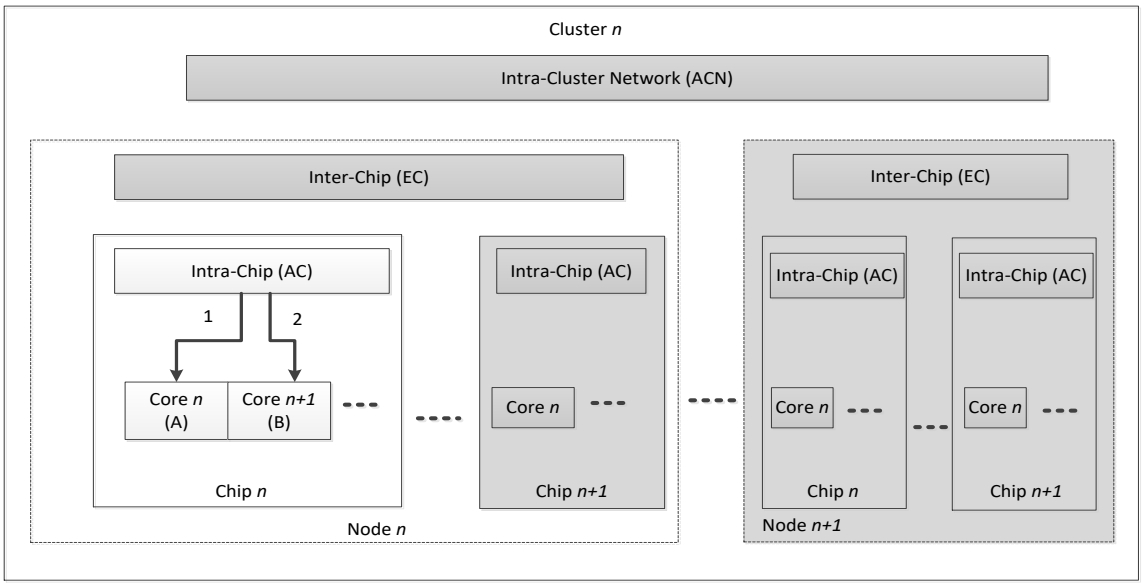

Figure 5: Communication for intra-chip network

Figure 6 shows inter-chip network (EC), following the sequence in the figure, so that data and information can be passed from one core of a chip to another core of another chip in the same node.

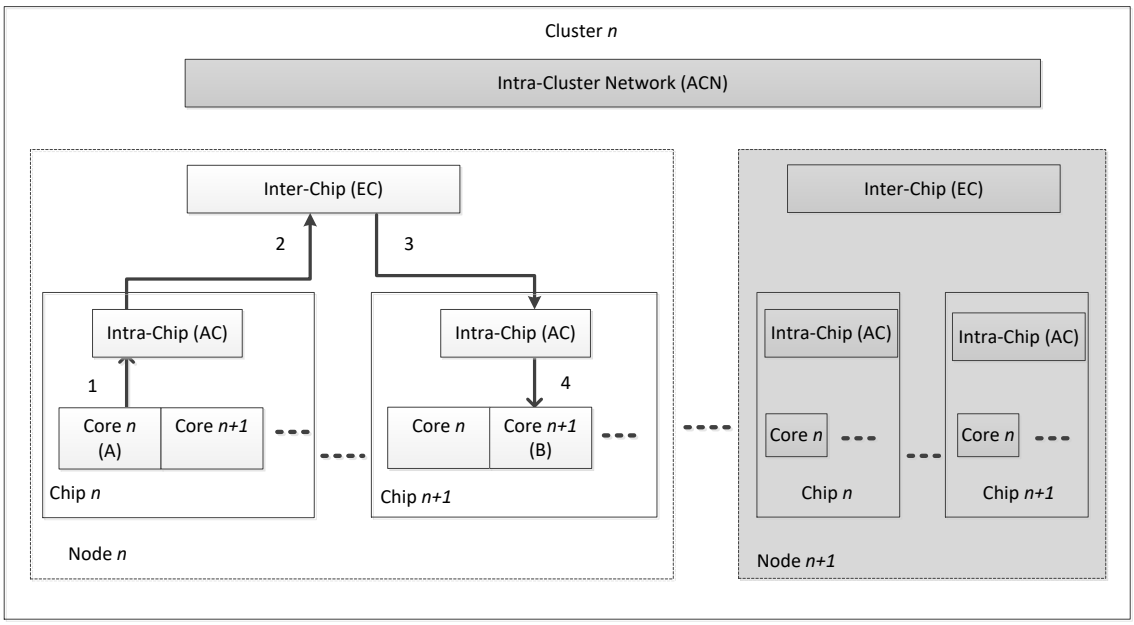

Figure 6: Communication for inter-chip network

Figure 7 shows the intra-cluster network (ACN) to connect nodes in a cluster and the connection is presented by the sequential number. It starts from the core, to the intra-chip, to inter-chip, to intra-cluster network and ten to the inter-chip of another node. Eventually all nodes can be connected.

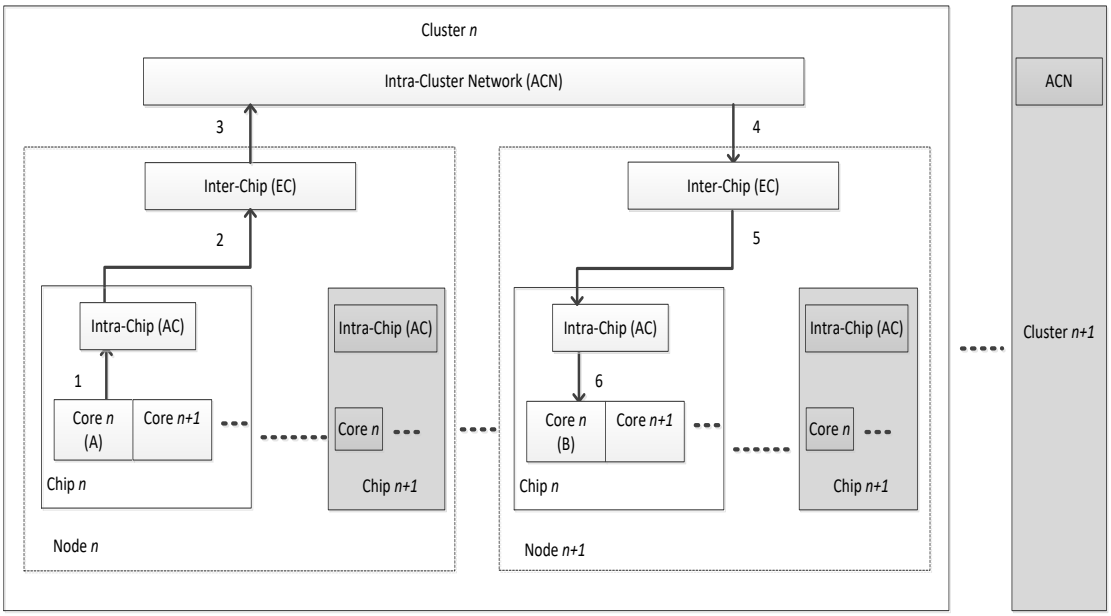

Figure 7: Communication routes for intra-cluster network

The next level is to illustrate inter-cluster network (ECN) and multi-cluster network (MCN), as shown in Figure 8. ECN can be used to transfer information and data between clusters, which can be connected to one another via the multi-cluster network (MCN). Following the sequence in the diagram, information can be passed from one core form a node to another core of another node in a different cluster. In this way, reliability and resiliency of communications can be maintained.

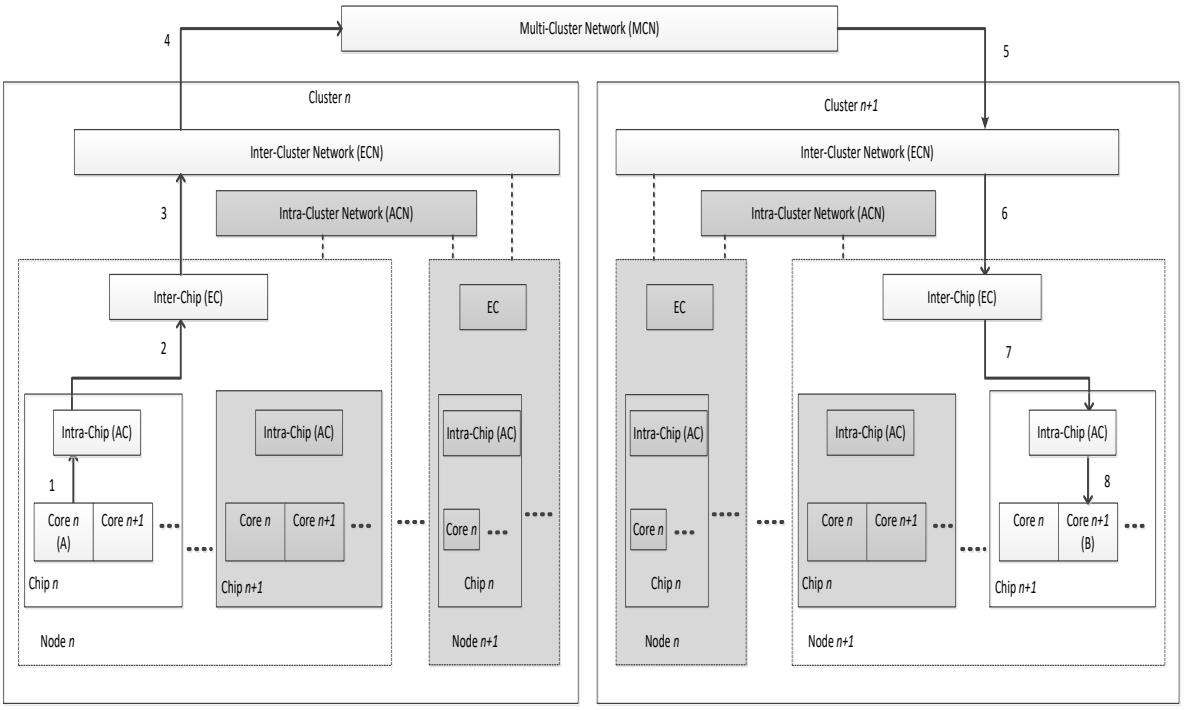

Figure 8: Communication routes for transmitting messages between clusters

3.2 MCAHS Queuing Network Model

The next section is to show the MCAHS Queuing Model. In interconnection networks, packets spend a lot of time waiting in queues before they are transmitted by a processor core to their destination. A source will generate packets at a rate of $\frac{1}{\lambda}$ packets per second and the packets will be in a queue while waiting to be transmitted into the network. An interconnection network then removes the packets from the queue on a first-in-first-out (FIFO) basis and processes them with an average transmission time [10].

'M/G/1 queuing networks' are used to analyse systems with a Poisson distribution transmission time [11-12]. The M/G/1 queuing network studies have been widely reported, which makes tractable the solution of modelling interconnection networks of MCAHS by simulation [11-13]. In general, an M/G/1 queuing network with arbitrary transmission time distribution has occupancy of –

$$W = \frac{(\beta)^2 \lambda_i}{2(1 - \beta\lambda_i)}$$

Where
λ_i = arrival rate
β = average transmission time

A traditional cluster contains single processor nodes with one interconnection network, and can be presented in Figure 8. Information passing between processors in single clusters can go through an intra-cluster network (ACN) which involves queues for messages to enter the network. Queuing networks for multi-core clusters are shown in Figure 9. Multi-core clusters are also included in single cluster architecture, but with multiple cores in a processor. With multiple cores in a chip, the combination may be able to provide greater throughput by reducing the queues in each processor [14]. This will decrease the latency and improve the interconnection network performance.

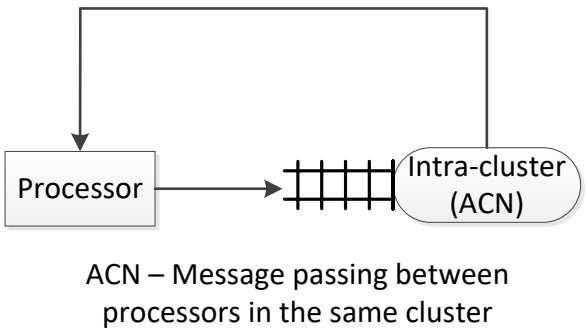

Figure 9: Queuing network of single-core cluster

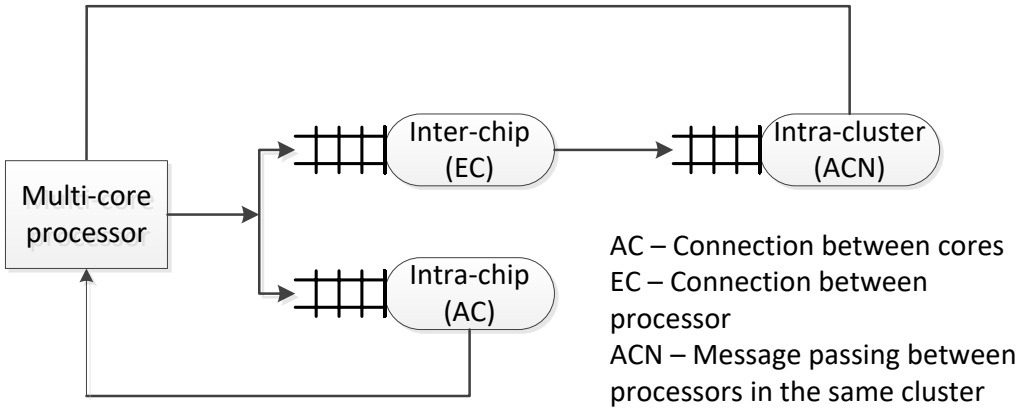

Figure 10: Queuing network of multi-core cluster

To demonstrate a similar concept to Figure 11, combining both Figure 9 and 10 will be a sensible approach to illustrate a good connection between Multi-core architecture for a HIS system (MCAHS). Compared to traditional clusters, multi-core clusters involve with three interconnection networks. Chip communication consists of intra-chip networks (AC) and inter-chip networks (EC), while communication between processors in the single cluster is via intra-cluster networks (ACN). Figure 11 shows queuing network of MCAH) to allow the multi-core processor to connect to any destination to get a reduced time and better performance.

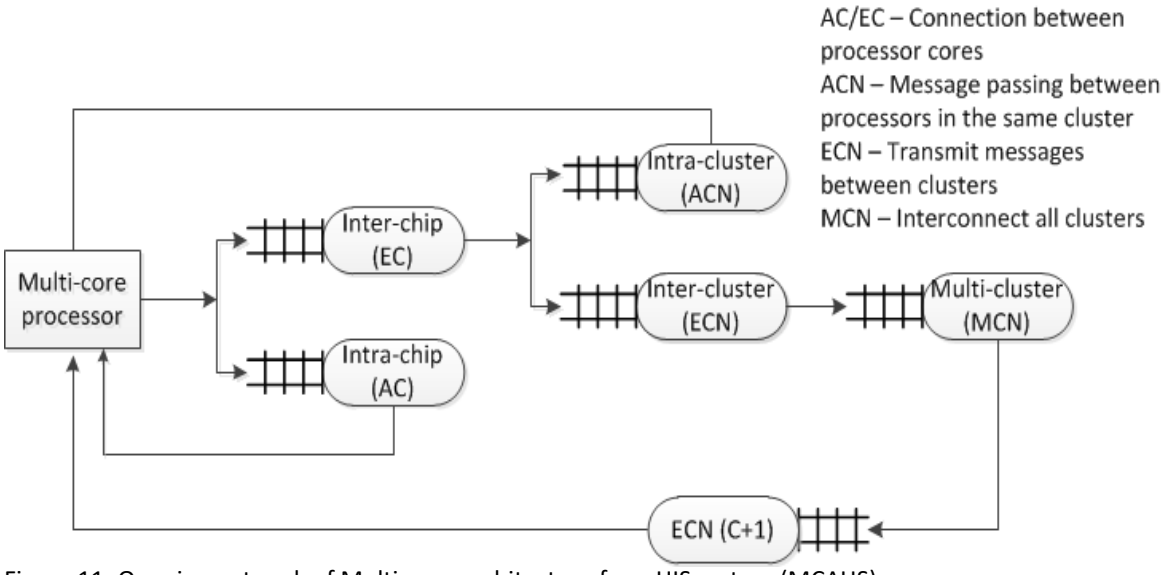

Figure 11: Queuing network of Multi-core architecture for a HIS system (MCAHS)

Figure 12 shows a flow diagram representing the work flow in a cluster node with a multi-core processor. It will check the status of the node is not idle before the next action. The target node will communicate with nodes through the interconnection network. If the status is idle, then more checks will be identified to ensure that before any major task, no network latency or idle state can cause further delay.

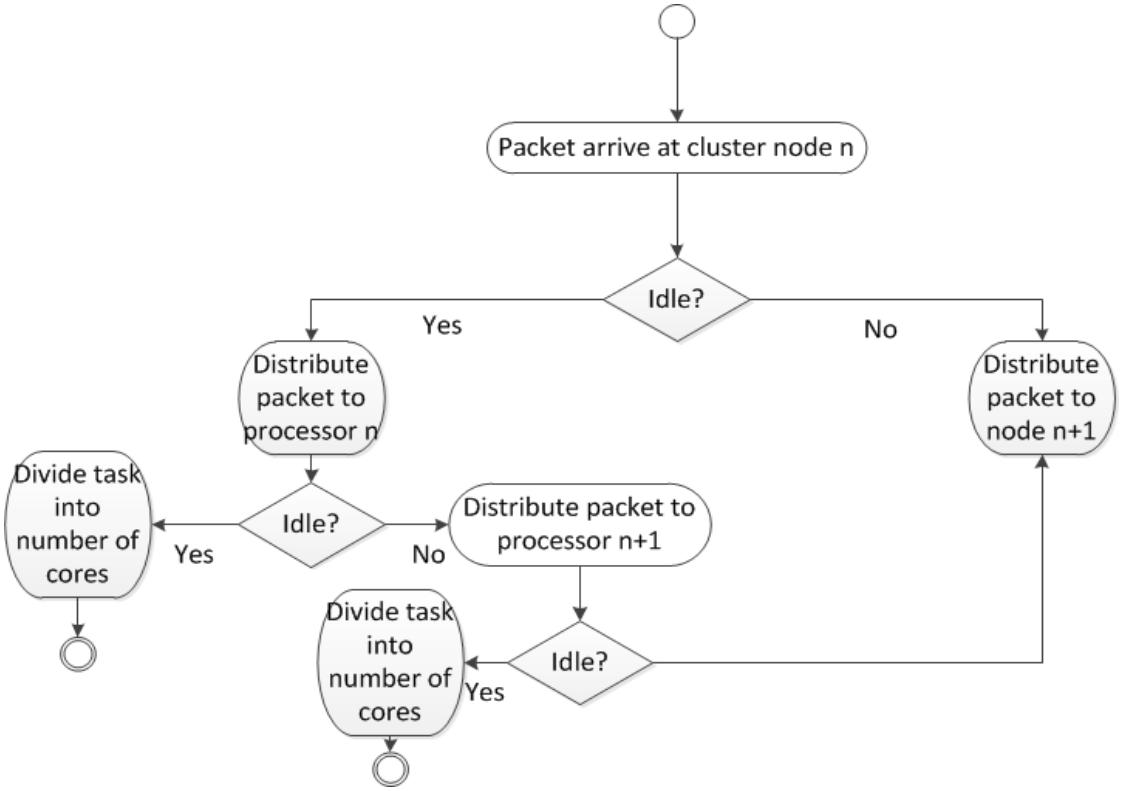

Figure 12: Packet flow in the cluster node of a multi-core processor

4. Experiments

This section presents experiments with multi-core clusters with different types of performance evaluation based on our HIS design. The aim is to investigate the average message latency (unit) versus traffic generation rate, number of clusters and message size, which will be part of criteria to QRR of a HIS system. Parameter II indicates key parameters (in their tables) to run experiments of Multi-core architecture for a HIS system (MCAHS).

4.1 Experiments with Multi-core Clusters

This section presents multi-core cluster experimental results for MCAHS. A simulation experiment was performed based on model cases in Table 1. Two different flow control methods, store-and-forward and wormhole, are used to validate the simulation model, whereby Figure 13 and 14 show the average message latency for both. Multi-core has a better performance since it can process data faster.

Table 1: Model cases for multi-core clusters

Items	Quantity
No. of cores *(nc)*	1, 2, 4
Message Length *(M)* and Flit Length *(F)*	32 flits, 256 bytes
No. of cluster, *m*-port, *n*-tree	8, 8, 2

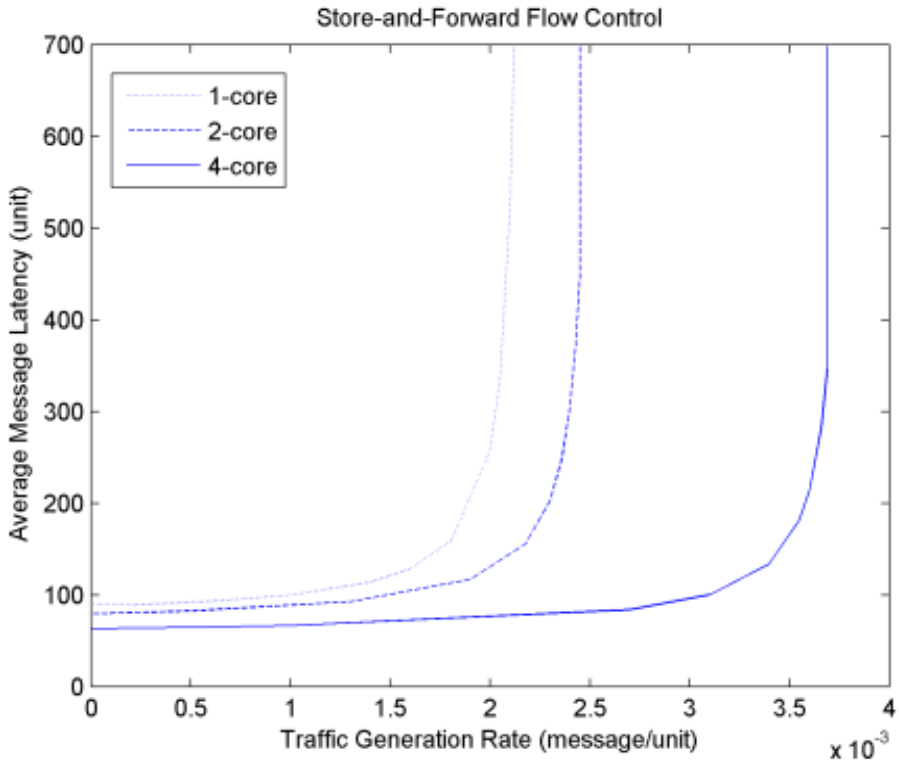

Figure 13: Average Message Latency based on Store-and-Forward Flow Control

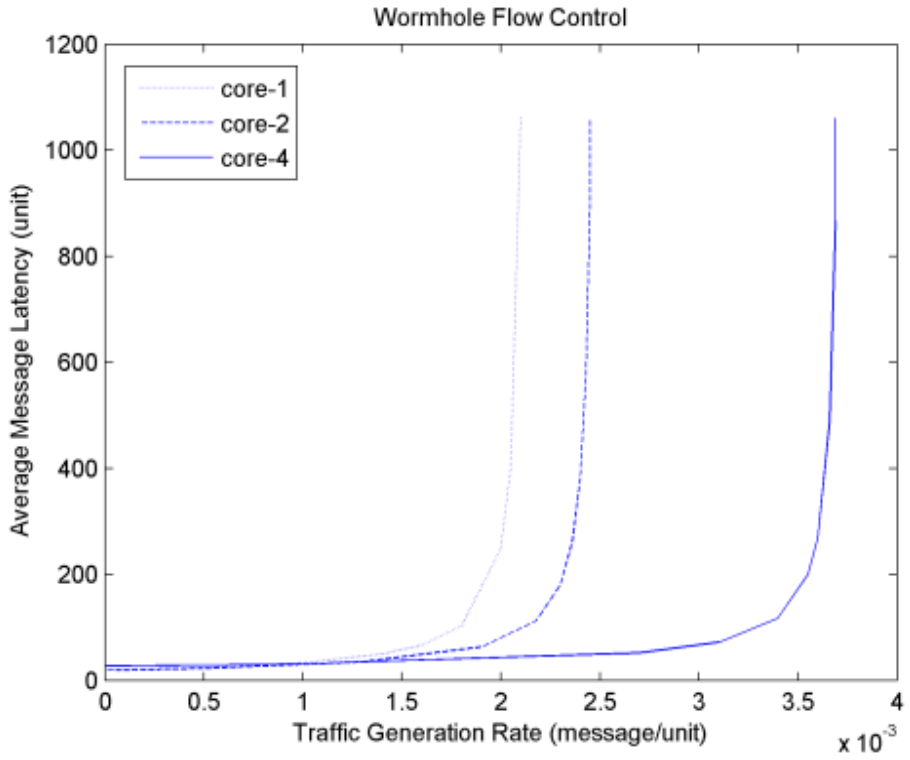

Figure 14: Average Message Latency based on Wormhole Flow Control

The impact on cluster size

This section presents three key parameters for network latency, switch latency and network bandwidth:

- For the internal-cluster, it has 0.02 s, 0.01s and 800 b/s.
- For the external-cluster, it has 0.01 s, 0.05s and 600 b/s

Table 2: Simulation Input set 1

Items	Quantity
No. of cluster *(C)*	8, 16, 32, 64, 128
No. of cores *(nc)*	1, 2, 4
Message generation rate *(λg)*	0.002s
Message Length *(M)*	8K
No. of *m*-port *n*-tree	4, 2

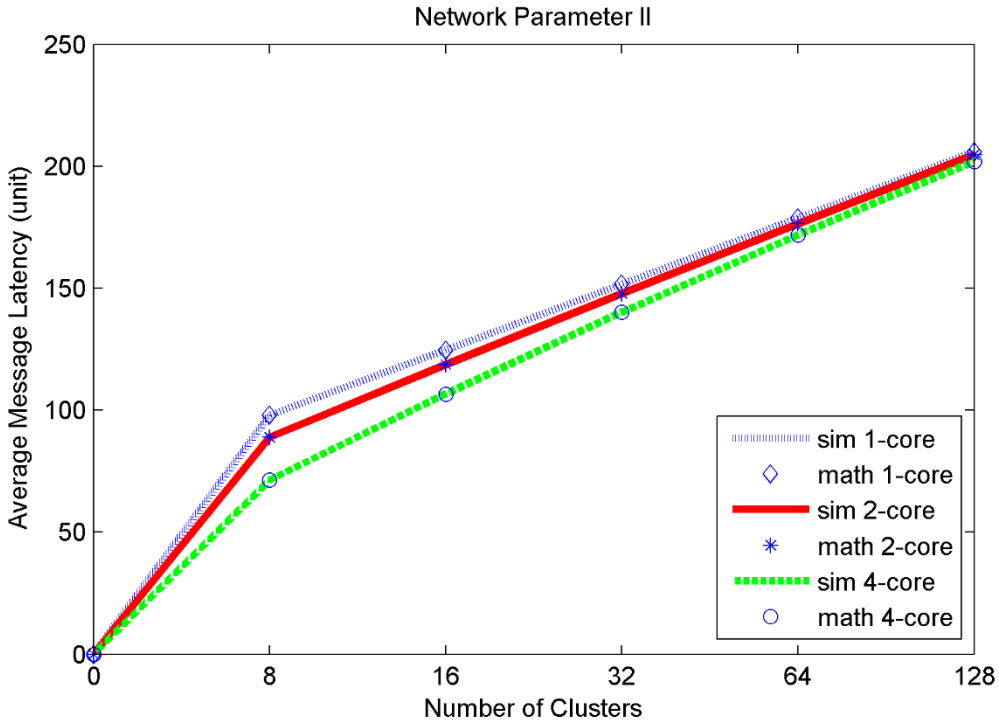

Figure 15: Average Message Latency vs. Cluster Size based on network parameter II

As shown in Figure 15, the average message latency increased while the number of clusters increased and experienced almost the same latency rate when at a larger cluster size. The saturation of the throughput also increased with the larger number of clusters. The results also indicate that, even with a larger cluster, Multi-core architecture for a HIS system (MCAHS) can save more transmission time and can finish the same tasks at a lower traffic rate.

What is observed in these experiments is important, as it reveals that the HIS can be used with various cluster sizes, including the traditional single-core cluster to clusters of larger size. With MCAHS, the capacity of the resources increase, so that more packets can be transmitted while experiencing lower latency.

4.2 The impact on message length and scalability

In this experiment, to examine the potential scalability in the cluster architecture, different message lengths were run, as reflected in Table 3.

Table 3: Simulation Input set 2

Items	Quantity
No. of cores *(nc)*	1, 2, 4
Message generation rate *(λg)*	0.001s
Message Length *(M/bytes)*	128, 256, 512, 1K, 2K, 4K, 8K, 16K
No. of cluster, *m*-port *n*-tree	8, 8, 2

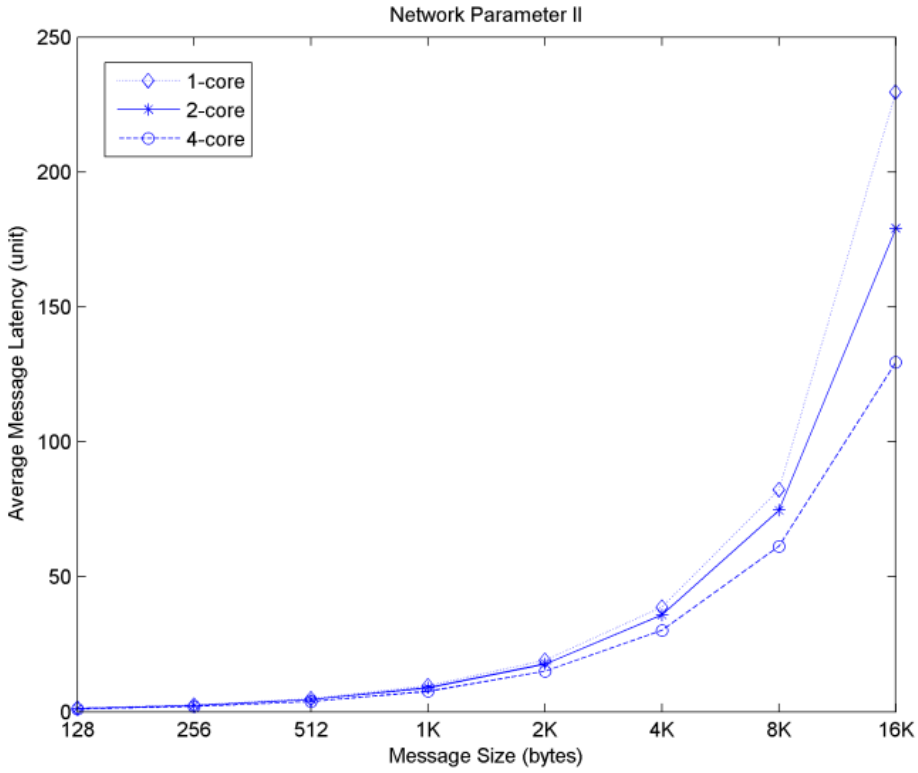

Figure 16: Average Message latency vs message size based on network parameter II

Figure 16 shows the average message latency based on various message sizes. The message sizes for the experiment range from 128 bytes to 16K as the largest message size. With the same message generation rate, 0.001s, the results reflected in both figures demonstrated that network latency happened more sharply at the 8K size of message for all cores, despite 2-core and 4-core processors having a less latency compared to the single-core processor. Even when the message sizes were simulated with different bandwidths, the latency

increased as the message sizes increased. With a smaller message size, the message latency increments for all cores were very small and almost similar. The significant differences start to occur at a message size of 1K and became obvious at the larger message sizes. This indicates that the architecture is scalable with different sizes of message.

4.3 Discussion from experiments

This section sums up analysis of experiments as follows. First, the latency experimental results suggest that multi-core processors can improve network performance by 51-76% compared to single-core processors. This indicates that optimizing all levels of interconnection network is important in this architecture. As the evaluation is based on store-and-forward flow control in Figure 11, the probability of blocking is zero, which contributes to higher saturation throughput.

Other experiments were conducted with various sizes of cluster. The architecture can scale well with small to larger sizes of cluster while achieving lower latency and higher throughput. Thus, these results can validate our HIS system having a good quality, reliability and resiliency (QRR). Experiments need to focus on “micro” level to ensure that even slightest changes, such as 8K message size, can impact on network latency. The results have reveals that small latency happens with smaller messages size but the latency increase with the larger message size. The experiments also demonstrated that MCAHS can scale well compared to traditional single-core cluster. Additionally, work in [15-18] show the importance of system design, network communications and security to ensure all the work can be safely, reliably and accurately be completed. Furthermore, proposal in [19-20] demonstrate importance of algorithm and smart model to achieve QRR for all the services.

5. Conclusion and Future Work

A MCAHS has been proposed and demonstrated to justify our research contributions for HIS system as follows. First, we present the system design from the chip to the clusters, and the design can ensure good information to be reached at its destination at the most convenient way. Different network queuing diagrams have been explained. Second, our experiments show a better latency management for multi-core systems and clusters. We also identify message size of 8K and below can be optimum for sending and processing large number of information and data. By reducing latency and ensuring good traffic within multi-core systems, performance can be good and results can be reliable. This can support the requirement of quality, reliability and resiliency of recommended HIS systems. Our work is relevant and contributing to the following HIS areas:

- Traffic analysis and management: Causes of latency have been identified and traffic can be better managed since latency can be reduced. Situations with increased latency can be avoided.
- Performance analysis of transmission control: Experiments on the latency under Store-and-Forward Flow Control method, Wormhole flow control method, cluster size and message size have been undertaken with analysis presented.

To demonstrate QRR, our emphasis was on latency – understand how it happened, what triggered its increase, and performed experiments with different types, including under Store-and-Forward Flow Control method, Wormhole flow control method, cluster size and message size to get a deep understanding. In this way, we could reduce latency and avoided its sharp increase. We justified our research contributions, particularly in the area of "traffic analysis and management" and "performance analysis of transmission control" of the HIS systems. Our future work will include working and integrating with big data, internet of things (IoT), deep learning and other pioneering systems to ensure QRR in all new services on offers.

6. References

[1] National Research Council. (2011). The Future of Computing Performance: Game Over or Next Level?. National Academies Press.

[2] R. Buyya, S. K., Garg, & R. N., Calheiros. (2011, December). SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In Cloud and Service Computing (CSC), 2011 International Conference on (pp. 1-10). IEEE.

[3] Z. A., Al-Hemyari, & A. M., Al-Sarmi. (2014). Features of performance indicators in quality improvement of HEIs. In Proceedings of the Ireland International Conference in Education (IICE), Oct (pp. 26-28).

[4] J., Sherwani, N., Ali, N., Lotia, Z., Hayat, & R., Buyya. (2004). Libra: a computational economy based job scheduling system for clusters. Software: Practice and Experience, 34(6), 573-590.

[5] J. A., Brown, R., Kumar, & D. Tullsen. (2007, June). Proximity-aware directory-based coherence for multi-core processor architectures. In Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures (pp. 126-134). ACM.

[6] M., Katevenis, S., Sidiropoulos, & C., Courcoubetis. (1991). Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. IEEE Journal on selected Areas in Communications, 9(8), 1265-1279.

[7] H. S., Wang, X., Zhu, L. S., Peh, & S., Malik. (2002). Orion: a power-performance simulator for interconnection networks. In Microarchitecture, 2002.(MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on (pp. 294-305). IEEE.

[8] V., Chang, & G., Wills. (2016). A model to compare cloud and non-cloud storage of Big Data. Future Generation Computer Systems, 57, 56-76.

[9] V., Chang. (2015). Towards a Big Data system disaster recovery in a Private Cloud. Ad Hoc Networks, 35, 65-82.

[10] D., Wentzlaff, P., Griffin, H., Hoffmann, L., Bao, B., Edwards, C., Ramey, ... & A., Agarwal. (2007). On-chip interconnection architecture of the tile processor. IEEE micro, 27(5), 15-31.

[11] T. G., Robertazzi. (2012). Computer networks and systems: queueing theory and performance evaluation. Springer Science & Business Media.

[12] H., Sharifi, M. K., Akbari & B., Javadi. (2009, August). An analytical model of communication networks in multi-cluster systems in the presence of non-uniform traffic. In Computational Science and Engineering, 2009. CSE'09. International Conference on (Vol. 1, pp. 79-86). IEEE.

[13] H., Sarbazi-Azad, M., Ould-Khaoua, & A. Y., Zomaya. (2005). Design and performance of networks for super-, cluster-, and grid-computing: Part I. Journal of Parallel and Distributed Computing, 65(10), 1119-1122.

[14] D., Geer. (2007). For programmers, multicore chips mean multiple challenges. Computer, 40(9).

[15] P., Vijayakumar, S. M., Ganesh, L. J., Deborah, & B. S., Rawal. (2016). A new SmartSMS protocol for secure SMS communication in m-health environment. Computers & Electrical Engineering.

[16] P., Vijayakumar, P., Pandiaraja, M., Karuppiah. & L. J., Deborah. (2017). An efficient secure communication for healthcare system using wearable devices. Computers & Electrical Engineering. 63, 232-245.

[17] F., Wei, P., Vijayakumar, J., Shen, R., Zhang, & L., Li. (2017). A provably secure password-based anonymous authentication scheme for wireless body area networks. Computers & Electrical Engineering.

[18] Namasudra, S., Roy, P., Vijayakumar, P., Audithan, S., & Balusamy, B. (2017). Time efficient secure DNA based access control model for cloud computing environment. Future Generation Computer Systems, 73, 90-105.

[19] W., Lin, Z., Wu, L., Lin, A., Wen, & J., Li. (2017). An Ensemble Random Forest Algorithm for Insurance Big Data Analysis. IEEE Access, 5, 16568-16575.

[20] W., Wu, W., Lin. & Z., Peng. (2017). An intelligent power consumption model for virtual machines under CPU-intensive workload in cloud environment. Soft Computing, 21(19), 5755-5764.

Biography

Dr. Norhazlina Hamid completed her PhD in Computer Science at the University of Southampton in 2015. She is currently working for Malaysian Government. This paper is based the improvement of her thesis.

Dr. Victor Chang is an Associate Professor and Director of PhD and MRes at IBSS, Xi'an Jiaotong-Liverpool University, China. He has more than 130 peer-reviewed publications. He is leading expert in this area and has given 16 keynotes internationally. He won many awards, including Outstanding Young Scientist Award 2017. He founded two conferences: IoTBDS and COMPLEXIS and two workshops.

Dr. Robert John Walter was a Lecturer in Computer Science at the University of Southampton, UK. He is a program chair of IoTBDS 2016-2018. He has recently retired.

Dr. Gary Brian Wills is an Associate Professor in Computer Science at the University of Southampton, UK. He has more than 300 peer-reviewed publications. He has chaired several conferences. He is a program chair of IoTBDS 2016-2018. He has more than 45 completions of PhD students up-to-date.