Javadi, Bahman, Akbari, Mohammad K. and Abawajy, Jemal 2005, Performance analysis of heterogeneous multi-cluster systems*, in Proceedings of the 2005 international conference on parallel processing workshops*, IEEE Computer Society, Los Alamitos, Calif., pp. 493-500.

Performance Analysis of Heterogeneous Multi-Cluster Systems

Bahman Javadi
Amirkabir University of Technology
Computer Eng. and IT Department
P.O.Box 19514, Tehran, Iran
javadi@ce.aut.ac.ir

Mohammad K. Akbari
Amirkabir University of Technology
Computer Eng. and IT Department
P.O.Box 19514, Tehran, Iran
akbari@ce.aut.ac.ir

Jemal H. Abawajy
Deakin University
School of Information Technology
Geelong, VIC 3217, Australia
jemal@deakin.edu.au

Abstract

When building a cost-effective high-performance parallel processing system, a performance model is a useful tool for exploring the design space and examining various parameters. However, performance analysis in such systems has proven to be a challenging task that requires the innovative performance analysis tools and methods to keep up with the rapid evolution and ever increasing complexity of such systems. To this end, we propose an analytical model for heterogeneous multi-cluster systems. The model takes into account stochastic quantities as well as network heterogeneity in bandwidth and latency in each cluster. Also, blocking and non-blocking network architecture model is proposed and are used in performance analysis of the system. The message latency is used as the primary performance metric. The model is validated by constructing a set of simulators to simulate different types of clusters, and by comparing the modeled results with the simulated ones.

1. Introduction

Over the past few years, the predispositions in parallel processing system design and deployment have been concentrating on networked distributed systems such as commodity-based *cluster computing* [1] and *grid computing* [2] systems. These network-based systems have proven to be cost-effective parallel processing tools for solving many complex scientific, engineering and commercial applications as compared to the conventional supercomputing systems. To construct a high performance cluster as well as to orchestrate its processing power, evaluation of the performance of various design trade-offs is required.

This paper addresses the performance analysis problem for heterogeneous multi-cluster computing systems. The motivation for considering such systems is that multi-cluster systems are gaining more importance in practice [4, 11, 12] and a wide variety of parallel applications are being hosted on such systems as well [4, 5]. To this end, we present a new methodology that is based on Jackson queuing network to analytically evaluate the performance of heterogeneous multi-cluster systems. The model takes into account stochastic quantities as well as network heterogeneity in bandwidth and latency in each cluster. The message latency is used as the primary performance metric. The model is validated using simulation.

The rest of the paper is organized as follows. In Section 2, the problem statement and related work are discussed. In Section 3, we give a brief overview of the multi-cluster system model used in this paper. In Section 4, we describe the proposed performance model. The communication network model for blocking and non-blocking architecture is described in Section 5. In Section 6, we present the model validation experiments. Finally, Section 7 summarizes our findings and concludes the paper.

2. Problem Statement and Related Work

Most of existing cluster computing systems are mainly of a single cluster under the domain of one management [3]. However, advances in computational and communication technologies has made it economically feasible to conglomerate multiple clusters of heterogeneous networked resources leading to the development of large-scale distributed systems known as multi-cluster systems that is gaining momentum both in academic and commercial sectors [4, 11, 12].

Hence, the focus in this paper is on heterogeneous multi-cluster computing systems. However, building efficient and cost-effective high-performance cluster systems is not that simple as plugging in and setting up all components [18]. Factors such as heterogeneity make creating and marinating a robust high performance cluster computing infrastructures a

Proceedings of the 2005 International Conference on Parallel Processing Workshops (ICPPW'05)
1530-2016/05 $20.00 © 2005 IEEE

significant challenge. The problem even becomes complicated when multiple independent clusters are conglomerated into one large-scale system as in [11, 12]. Moreover, performance analysis in multi-cluster computing systems has proven to be a challenging task that requires the innovative performance analysis tools and methods to keep up with the rapid evolution and ever increasing complexity of such systems [21]. There are three possible ways to address this problem - simulation, prediction and analytical modeling.

Current research in performance analysis issues for cluster computing is mainly based on exhaustive simulations [4, 21]. Simulation appears to be the only feasible way to analyze algorithms on large-scale distributed systems of heterogeneous resources. Unlike using the real system in real time, simulation works well, without making the analysis mechanism unnecessarily complex, by avoiding the overhead of coordination of real resources. The limitations of simulation-based solutions are that it is highly time-consuming and expensive. Similarly, techniques based on predictions from measurements on existing clusters would be impractical [8].

An alternative to simulation and prediction approaches is an analytical model, which is the focus of this paper. An accurate analytical model can provide quick performance estimates and will be a valuable design tool. However, there is very little research addressing analytical model for heterogeneous multi-cluster systems. The few results that exist are based on homogenous cluster systems and the evaluations are confined to a single cluster [8, 9, 10]. With all probability, multiple cluster systems would be configured from heterogeneous components, rendering exiting optimization solutions unusable in heterogeneous multi-cluster environment. In contrast, our work focuses on heterogeneous multi-cluster computing systems. To this end, we present a generic model to analytically evaluate the performance of multi-cluster systems. We believe that our work is the first to deal with heterogeneous multi-cluster environments.

3. System Model

Generally, multi-cluster systems can be classified into *Super-Cluster* and *Cluster-of-Cluster*. A good example of Super-Cluster systems is DAS-2 [11], which is characterized by large number of homogenous processors and heterogeneity in communication networks. In contrast, Cluster-of-Clusters are constructed by interconnecting multiple single cluster systems thus heterogeneity may be observed in communication networks as well as processors. The LLNL multi-cluster system which is built in by interconnecting of four single clusters, MCR[1], ALC[2], Thunder, and PVC[3] [12] is an example of cluster-of-cluster system.

Since we are planning to sketch a general model for multi-cluster systems, a generic structure of such systems is proposed. The new structure is called Heterogeneous Multi-Stage Clustered Structure (HMSCS), which is a derivative of MSCS[4] [7]. Figure 1 shows the overall architecture of HMSCS system. The system is made up of C clusters, each cluster i is composed of N_i processors of type T_i, $i=1,...,C$. Also, each cluster has two communication networks, an Intra-Communication Network ($ICN1_i$), which is used for the purpose of message passing between processors, and an intEr-Communication Network ($ECN1_i$), which is used to transmit messages between clusters, management and also for the expansion of system. Note that, ECN can be accessed directly by the processors of a cluster without going through the ICN. As it can be seen, this structure can cover two classes of multi-cluster systems.

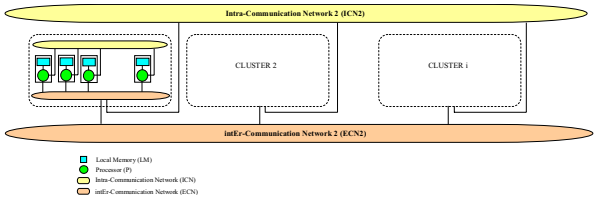

Figure 1 . Heterogeneous Multi-Stage Clustered Structure

4. Analytic Performance Modeling

In this section, to illustrate the derivation of the model, we will focus our discursion on the Super-Cluster system with homogenous processors and heterogeneous communication networks. At first, we should outline the assumptions made in the analysis, as following.

4.1. Assumptions

The proposed model is based on the following assumptions that are widely used in the similar study [7, 9, 10, 17, 20]:

[1] Multiprogrammatic Capability Cluster
[2] ASC Linux Cluster
[3] Visualization Cluster
[4] Multi-Stage Clustering Structure

1. Each processor generates packets independently which follows a Poisson process with a mean rate of λ and inter-arrival times are exponentially distributed.

2. The arrival process at a given communication network is approximated by an independent Poisson process. This approximation has often been invoked to determine the arrival process in store-and-forward networks [19]. In this paper we apply the store-and-forward network, e.g., Ethernet-based networks. Therefore, the rate of process arrival at a communication network can be calculated using Jackson's queuing networks formula [19].

3. The destination of each request would be any node in the system with uniform distribution.

4. The processors which are source of request must be waiting until they get service and they cannot generate any other request in wait state.

5. The number of processors in all clusters are equal ($N_0=N_1=N_2=...=N_C$) with homogenous type of ($T_0=T_1=T_2=...T_C$).

6. Message length is fixed and equal to M bytes.

4.2. Proposed Analytical Model

Based on characteristics of the HMSCS system behavior (see Figure 1) each communication network can be considered as service center. The queuing network model of system is shown in Figure 2, where the path of a packet through various queuing centers is illustrated. As is shown in the model, the processor requests will be directed to service center ICN1 and ECN1 by probability *1-P* and *P*, respectively. According to assumption 1, the request rate of a processor is λ, so the input rate of ICN1 and ECN1 which feed from that processor will be $\lambda(1-P)$ and $\lambda.P$, respectively. The additional inputs at these service centers, γ_{I1} and γ_{E1}, are due to the requests generated by other processors of the same cluster. The output of ICN1 is feedback to the same processor, and also ε_{I1} represents the response to other processors in the same cluster.

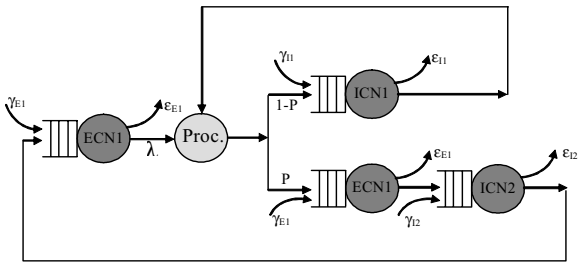

Figure 2 . Queuing Model of a SuperCluster System

The external request (out of cluster) of a cluster goes through the ECN1 with probability P and then ICN2. In the return path, it again accesses the ECN1 to get back to the node, which initiated the request. As mentioned before, εE1 and εI2 are responses to the other requests except the one under consideration. So, the total requests of the processors received by service centers in the first stage can be calculated as follows:

$$\lambda_{I1} = (1- P)\lambda + \gamma_{I1} = (1-P)\lambda + (N_0-1)(1-P)\lambda = N_0 (1-P)\lambda \quad (1)$$

$$\lambda_{E1(1)} = P\lambda + \gamma_{E1} = P\lambda + (N_0-1)P\lambda = N_0 P\lambda \quad (2)$$

where $\lambda_{E1(1)}$ is the input rate of ECN1, the one which is feed by the processor.

In the second stage, the input request rate of ICN2 in forward path and ECN1in feedback path can be computed by following equations:

$$\lambda_{I2} = \lambda_{E1(1)} + \gamma_{I2} = N_0 P\lambda + (C-1)\lambda_{E1(1)} = N_0 P\lambda + (C-1) N_0 P\lambda = C N_0 P\lambda \quad (3)$$

$$\lambda_{E1(2)} = \lambda_{I2}/C = N_0 P\lambda \quad (4)$$

where $\lambda_{E1(2)}$ is the input rate of ECN1 from feedback path. According to equations (2) and (4), the input rate of ECN1 is:

$$\lambda_{E1} = \lambda_{E1(1)} + \lambda_{E1(2)} = 2N_0 P\lambda \quad (5)$$

The average number of waiting processors in each service center can be computed through queue length of each center. So, the average of total waiting processors in the system will be:

$$L = C (2L_{E1} + L_{I1}) + L_{I2} \quad (6)$$

which L is denoting the queue length of each service center. As mentioned in the assumption 4, the waiting processors would not be able to generate new requests, so the effective request rate of the processor would be less than λ. Applying the method described in [13] to find the effective request rate of a processor, it is directly dependent to the ratio of number of active processors to total number of processors. Therefore, L and λ are computed iteratively based on following equation, until no considerable change is observed between two consecutive steps:

$$\lambda_{eff} = \frac{N-L}{N} \times \lambda \quad (7)$$

As it can be seen in the previous equations, the probability P has been used as the probability of outgoing request within a cluster. According to assumption 3, this parameter is computed base on structure of HMSCS by the following equation:

$$P = \frac{(C-1) \times N_0}{(C \times N_0) - 1} \quad (8)$$

In this paper, message latency is selected as a primary performance metric. However, most of the other performance metrics for the queuing network model of a multi-cluster system are related to the message latency with simple equations [17]. To model the mean message latency, we consider effective parameters as follows. In such systems, the mean network latency, that is the time to cross the network, is the most important part of the message latency. Other parameters such as protocol latency can be negligible.

Since the system under study is symmetric, averaging the network latencies seen by message generated by only one node for all other nodes gives the mean message latency in the network. Let S be the source node and D denotes a destination node such that $D \varepsilon A\text{-}\{S\}$ where A is the set of all nodes in the network. The network latency, T_C , seen by the message crossing from node S to node D consist of two parts: one is the delay due to the physical message transmission time, T_W, and the other is due to the blocking time in the network, T_B. Therefore, T_C can be written as:

$$T_C = T_W + T_B \quad (9)$$

These parameters are strongly depended on the characteristics of the communication network which is used in the system. Of this, we take into account two different networks in our model as following.

5. Modeling of Communication Network

The key requirements for speedup of clustered parallel applications are an interconnect that allows the cluster nodes to communicate with each other as quickly as possible. The most important characteristics for the cluster interconnect are as follows:

Latency: the time to transmit a small message on the network which is typically measured in microseconds (μs).

Bandwidth: the rate of throughput for large messages when are pipelined into interconnects fabric and is typically measured in megabytes/second (MB/s).

Bisection bandwidth: the rate of communication between two halves of the system which is measured in megabytes/second (MB/s).

The first two parameters are strongly influence by the network technologies i.e., Ethernet, Myrinet, Infiniband, etc. But the last one is related to the structure of the network, where interconnection networks of most regular computing systems are characterized by their bisection bandwidth. In this study, all above mentioned parameters play important role in our performance model of interconnection networks. Due to importance of last parameter, it will be discussed in detail next.

5.1. Bisection Width

More formal definition of a communication bottleneck is based on a property known as the *bisection width*, which is the minimum number of links that must be cut in order to divide the topology into two independent networks of the same size (plus or minus one node). For example, the bisection width of a tree is 1, since if either link connected to the root is removed the tree is split into two subtrees. The *bisection bandwidth* of a parallel system is the communication bandwidth across the links that are cut in defining the bisection width. This bandwidth is useful in defining worst case performance of algorithms on a particular network, since it is related to the cost of moving data from one side of the system to the other.

To see why the bisection width is an important characteristic, consider the following example: Sometimes, results calculated by one half of the network might be needed by the other half. If the bisection width of the network is b, which is much smaller than n, the network will spend n/b steps just shipping values around. A larger bisection width enables faster information exchange, and is, therefore, preferable.

Definition 1: *A network with N nodes has "Full Bisection Bandwidth" if the sum of the link bandwidths between any two halves of the network is N/2 of a single link bandwidth.*

In the following sections, analytical model for non-blocking and blocking interconnect architecture are articulated.

5.2. Non-blocking Network Model

Based on aforementioned parameters, for non-blocking interconnect architecture, the time to transmit a message of size M from/at node S with index of i

to/from node *D* with index of *j*, similar to [14], can be obtained from following formula:

$$T_{ij} = \alpha_{ij} + M.\beta_{ij} \qquad (10)$$

where α_{ij} is the network latency and β_{ij} is the time to transmit a byte (inverse of bandwidth). So, the network heterogeneity in our model was considered using the α_{ij} and β_{ij} values.

For non-blocking architecture, we use a Multi-Stage Fat-Tree topology which is used in some cluster systems such as Thunder [15], with some minor modifications. The building block in this topology is a *Pr*-way switch fabric. Where switch's ports are divided into Up-Link (*UL*) and Down-Link (*DL*) connections. In the middle stage we have *UL=DL=Pr/2*, but in the last stage, *DL* is equal to *Pr* and *UL* is zero. Figure 3 depicts this topology of 16 nodes connected through 8-port switches. Since this topology posses a multi-level switch, it causes the latency of the network to be increased. So the equation (10) can be rewritten as follows:

$$T_{ij} = \alpha_{ij} + (2d\text{-}1)\alpha_{sw} + M.\beta_{ij} \qquad (11)$$

where α_{sw} is the latency of a network switch and *d* is the number of stages in the topology and can be written as:

$$d = \left\lceil \frac{Log_2^N - 1}{Log_2^{\Pr} - 1} \right\rceil \qquad (12)$$

where *N* is number of nodes which are to be connected to the network.

Proposition 1: *the number of switches in a multi-stage fat-tree topology can be calculated as follows:*

Since each stage has $\left\lceil \frac{N}{DL} \right\rceil$ switches, so for all stages except the last one, the number of switches can be obtained from $\sum_{i=1}^{d-1}\left(\left\lceil \frac{N}{\Pr/2} \right\rceil\right) = \sum_{i=1}^{d-1}\left(\left\lceil \frac{2N}{\Pr} \right\rceil\right)$ and for the last stage, from $\left\lceil \frac{N}{\Pr} \right\rceil$. Therefore the total number of switches will be:

$$k = \sum_{i=1}^{d-1}\left(\left\lceil \frac{2N}{\Pr} \right\rceil\right) + \left\lceil \frac{N}{\Pr} \right\rceil$$

$$\Rightarrow k = (d-1)\times\left\lceil \frac{2N}{\Pr} \right\rceil + \left\lceil \frac{N}{\Pr} \right\rceil \qquad (13)$$

In Figure 3, for example the number of stages can be computed from equation (12) which is *d=2*, and from equation (13) the total number of switches will be *k=6*.

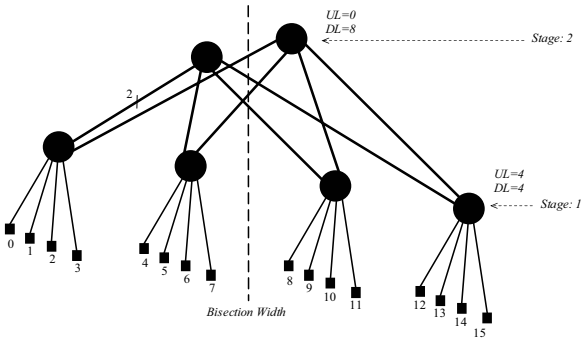

Figure 3. The Multi-Stage Fat-Tree (N=16, Pr=8)

Theorem 1: *A multi-stage fat-tree topology is a communication network with full bisection bandwidth.*

Proof. According to the proposition 1, the number of switches in stage-*d* is $\left\lceil \frac{N}{\Pr} \right\rceil$, so if we divide the network into two halves, the number of links from left half to the right one should be $\frac{1}{4}\times\left\lceil \frac{N}{\Pr} \right\rceil\times\Pr$, and we have the same number of links from right half to the left side. Therefore, bisection width can be calculated as follows:

$$\text{Bisection width} = \frac{1}{4}\times\left\lceil \frac{N}{\Pr} \right\rceil\times\Pr + \frac{1}{4}\times\left\lceil \frac{N}{\Pr} \right\rceil\times\Pr$$

$$= \frac{1}{2}\times\left\lceil \frac{N}{\Pr} \right\rceil\times\Pr = \frac{1}{2}\times\lceil N \rceil = \frac{N}{2} \qquad (14)$$

As a result, the bisection bandwidth is *N/2* times of a single link bandwidth, and according to the definition 1 this topology is a communication network with full bisection bandwidth. €

It is obvious that in this architecture the network bandwidth doesn't have any degradation at all. Therefore, blocking time in the network is equal to zero ($T_B = 0$) and consequently $T_C = T_W$. Now, with assumption of exponential distribution for service time of the communication networks, we can determine the mean transmission time (T_W). Due to non-blocking behavior of the network, it seems which this is a reasonable assumption. Of this and according to Figure 2, we can write:

$$T_W = (1\text{-}P)W_{I1} + P(W_{I2} + 2W_{E1}) \qquad (15)$$

where W_i is the waiting time of each service center and can be computed as in bellow:

$$W_i = \frac{1}{\mu_i - \lambda_i} \qquad (16)$$

5.3. Blocking Network Model

Here, we consider a blocking interconnect architecture, to propose an analytical model similar to what happened in non-blocking network model. To

construct such networks, a chain of switches to be need cascaded to each other, and despite non-blocking network, one level switch is used here. In other words, this topology is a *Linear Array* of switches, in which the number of switches in this network can be calculated as follows (having *d=1* in the equation (13)):

$$k = \left\lceil \frac{N}{\Pr} \right\rceil \qquad (17)$$

So, the time to transmit a message of size *M* from/at the node *i* to/from the node *j*, can be calculated as:

$$T_{ij} = \alpha_{ij} + \varphi.\alpha_{sw} + M.\beta_{ij} \qquad (18)$$

where φ is the number of a network switch which is traversed form node *S* to node *D* ($1 \le \varphi \le k$). On the basis of average case analysis, the φ can be substituted with the average of traversed distances in the network (a linear array topology). So the equation (18) can be written as:

$$T_{ij} = \alpha_{ij} + \frac{k+1}{3}.\alpha_{sw} + M.\beta_{ij} \qquad (19)$$

Since our topology is a linear array of switches, it is obvious that the bisection width of this topology must be one, so the bisection bandwidth is equal to a single link bandwidth. According to the definition 1 this topology is not a full bisection bandwidth, as a result $T_B \neq 0$. Due to assumption 4, having completely uniform network traffic, the probability of crossing a message between two halves is equal ½. Since the network throughput (allocated one) depends on the *aggregate bandwidth*, the number of communication pairs and the volume of the communication [18], thus (*N/2-1*) nodes will be blocked if all nodes request to transmit a message while only one is permitted to go through. Consequently, the blocking time can be given as:

$$T_B = (\frac{N}{2} - 1) \times M.\beta_{ij} \qquad (20)$$

The concept of bisection bandwidth confirms that the linear array network is not suited for random traffic patterns, but for localized traffic patterns. To calculate the total message latency, we make an approximation to simplify the model. To do this, we add the blocking time to the average transmission time of messages (equation (19)) and assume that the service time of the communication network has exponential distribution. So,

$$T_{ij} = \alpha_{ij} + \frac{k+1}{3}.\alpha_{sw} + M.\beta_{ij} + (\frac{N}{2} - 1) \times M.\beta_{ij}$$
$$= \alpha_{ij} + \frac{k+1}{3}.\alpha_{sw} + \frac{N}{2}.M.\beta_{ij} \qquad (21)$$

This means that the network throughput of this topology is slashed by number of nodes of one half. Now, similar to non-blocking network model the *Tc* can be calculated with equation (15).

6. Model Validation

In order to validate the technique and justify the approximations, the model was simulated. Requests are generated randomly by each processor with an exponential distribution of inter-arrival time with a mean of $1/\lambda$. The destination node (*D*) is determined by using a uniform random number generator. Each packet is time-stamped after its generation. The request completion time is checked in to compute the message latency in a "*sink*" module. For each simulation experiment, statistics were gathered for a total number of 10,000 messages.

Two different communication network scenarios for network heterogeneity were investigated which is listed in Table 1. In our study, we applied two well-known network technologies, Gigabit Ethernet (GE) and Fast Ethernet (FE), which are widely used in cluster systems. The other assumptions regarding to the model specifications and parameters are indicated in Table 2. It should be noted that the latency and bandwidth of each network are reported by [16] and our experimentation tests. The message generation rate (λ) is equal to 0.25 msg./sec in all experiments.

Table 1. Two Scenarios of Communication Networks

Cases	ICN1	ECN1 and ICN2
Case 1	Gigabit Ethernet	Fast Ethernet
Case 2	Fast Ethernet	Gigabit Ethernet

Table 2. Model Parameters

Items	Quantity	Unit
GE Latency	80	μs
GE Bandwidth	94	MB/s
FE Latency	50	μs
FE Bandwidth	10.5	MB/s
# of Ports in Switch Fabric (*Pr*)	24	Port
Switch Latency	10	μs
Msg. Generation rate (λ)	0.25	/s

Some combinations of system configuration, network type and message length were examined. In this regards, a multi-cluster system with *N=256* nodes and non-blocking communication network was selected as a platform to calculate the average message latency. The results of simulation and analysis are depicted in Figure 4 and Figure 5, in which the average message latencies are plotted against the number of clusters of the system with message sizes of 1024 and 512 bytes.

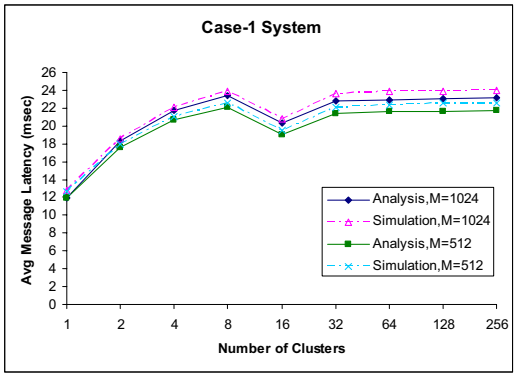

Figure 4. Average Message Latency vs. Number of Clusters for Non-blocking Networks in Case-1

As it can be seen in the figures, when the number of clusters is equal to 16, we experience a different behavior of the message latency which is due to usage of one switch fabric for all communication networks in the system. The reason is that the number of clusters and also the number of nodes in each cluster are less than number of ports in each switch fabric.

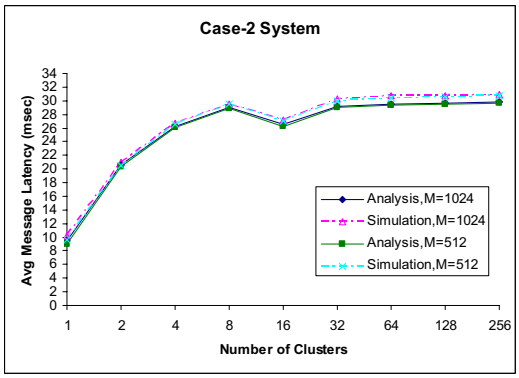

Figure 5. Average Message Latency vs. Number of Clusters for Non-blocking Networks in Case-2

In the next step, we moved to blocking communication networks (with the same parameters) to validate our model. As it was expected, here the average message latency was much larger than the previous model (non-blocking), which is depicted in Figure 6 and Figure 7. These figures demonstrate the average message latency in blocking network with uniform traffic pattern. Comparing with non-blocking network results, the average message latency of blocking network is larger, something between 1.4 to 3.1 times.

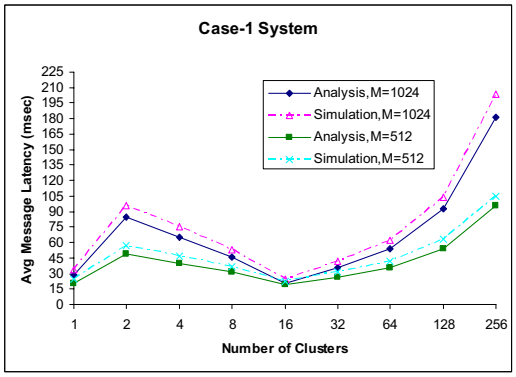

Figure 6. Average Message Latency vs. Number of Clusters for Blocking Networks in Case-1

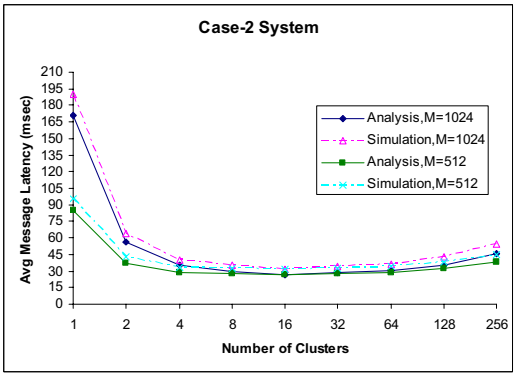

Figure 7. Average Message Latency vs. Number of Clusters for Blocking Networks in Case-2

The results of this study showed that our analytical model can predict the average message latency with good degree of accuracy.

7. Conclusions

A performance model is an essential tool for behavior prediction of a system. It is used to analyze intricate details of the system and various design optimization issues. One such model based on queuing networks is presented in this study to predict the message latency of multi-cluster systems. Two different networks, blocking and non-blocking, were used in our modeling of the system. The analysis captures the effect of communication network architecture on the system performance. The model is validated by constructing a set of simulators to simulate different types of clusters, and by comparing the modeled results with the simulated ones.

The future works focus on improving the analytical model to tack into account more effective parameters, modeling of communication networks with technology

heterogeneity and propose a similar model to another class of multi-cluster systems, Cluster-of-Clusters.

Acknowledgment: We would like to thank A.Jalalzadeh and Dr. S. Khorsandi for remarks and discussions. The help of Maliha Omer is also greatly appreciated.

8. References

[1] M. Q. Xu. "Effective meta-computing using LSF multicluster", In *Proceedings of CCGrid'2001*, pp. 100-106, May 2001, Brisbane, Australia.

[2] I. Foster, "The grid: A new infrastructure for 21st century science", *Physics Today*, pp. 42-47, 2002.

[3] A. V. Stergios and K. C. Sevcik, "Parallel application scheduling on networks of workstations", *Journal of Parallel and Distributed Computing*, 43(1):1159-1166, 1997.

[4] J. H. Abawajy and S. P. Dandamudi, "Parallel Job Scheduling on Multi-Cluster Computing Systems," In P*roceedings of the IEEE international Conference on Cluster Computing (CLUSTER'03)*, Dec. 1-4, 2003, Hong Kong.

[5] A.I. D. Bucur and D.H. J. Epema, "The influence of the structure and sizes of the jobs on the performance of co-allocation", In *Proceedings of ISSPP*, pp. 154-173, 2000.

[6] R. van Nieuwpoort, J. Maasen, H.E. Bal, T. Kielmann, and R. Veldema, "Wide-area parallel programming using the remote method invocation model", *Concurrency: Practice and Experience*, pp. 643-666, 2000.

[7] H. S. Shahhoseini, "Structural Design and Modeling of the Multistage Clustering Parallel Processing System," PhD Dissertation, Iran University of Science and Technology, March 1999.

[8] X. Du, X. Zhang, Z. Zhu, "Memory Hierarchy Consideration for Cost-Effective Cluster Computing," *IEEE Transaction on Computers*, Vol. 49, No.5, pp. 915-933, Sept. 2000.

[9] B. Javadi, S. Khorsandi, and M. K. Akbari, "Study of Cluster-based Parallel Systems using Analytical Modeling and Simulation" *International Conference on Computer Science and its Applications (ICCSA 2004)*, May 2004, Perugia, Italy.

[10] B. Javadi, S. Khorsandi, and M. K. Akbari, "Queuing Network Modeling of a Cluster-based Parallel Systems", *7th International Conference on High Performance Computing and Grids (HPC ASIA 2004)*, July 2004, Tokyo, Japan.

[11] The DAS-2 Supercomputer. http://www.cs.vu.nl/das2

[12] B. Boas, "Storage on the Lunatic Fringe", Lawrence Livermore National Laboratory, Panel at SC2003, Nov. 2003, Arizona, USA.

[13] H. S. Shahhoseini, M. Naderi, "Design Trade off on Shared Memory Clustered Massively Parallel Processing Systems", *The 10th International Conference on Computing and Information (ICCI '2000),* Nov. 2000, Kuwait.

[14] Y. Yan, X. Zhang, Y. Song, "An effective and practical performance prediction model for parallel computing on non-dedicated heterogeneous NOW", *J. Parallel and Distributed Computing*, pp. 63-80, 1996.

[15] "Thunder Statement of Work", University of California, Lawrence Livermore National Laboratory, Sept. 2003.

[16] M. Lobosco, and L. de Amorim, "Performance Evaluation of Fast Ethernet, Giganet and Myrinet on a Cluster", *Lecture Notes in Computer Science*, volume 2329, pp. 296-305, 2002.

[17] B. Javadi, M. K. Akbari, J.H. Abawajy, "Performance Analysis of Multi-Cluster Systems Using Analytical Modeling", *International Conference on Modeling, Simulation and Applied Optimization*, Sharjah, United Arab Emirates, Feb. 2005.

[18] A.T.T. Chun, Performance Studies of High-Speed Communication on Commodity Cluster, *PhD Dissertation*, University of Hong Kong, Dec. 2001.

[19] L. Kleinrock, *Queuing System: Computer Applications*, Part 2, John Wily Publisher, New York, 1975.

[20] H. Sarbazi-Azad, A. Khonsari, M. Ould-Khaoua, "Analysis of k-ary n-cubes with Dimension-order Routing", *Journal of Future Generation Computer Systems*, pp. 493-502, 2003.

[21] J. H. Abawajy, "Dynamic Parallel Job Scheduling in Multi-cluster Computing Systems," *4th International Conference on Computational Science*, Kraków, Poland, pp. 27-34, 2004.