A Queueing Analysis for Job Assignment on Two-Type Heterogeneous Supercomputer System

C.H. Wang
Department of Industrial and Information Management
National Cheng Kung University
Taiwan

Y.T. Chen
Supercomputing Research Center, Computer and Network Center
National Cheng Kung University
Taiwan

C.C. Hwang
Department of Engineering Science
National Cheng Kung University
Taiwan

***Abstract*—This paper introduces a queueing analysis of a two-tier service supercomputer system, where one type of processors offers service with a finite waiting buffer and the other offers unlimited waiting buffer. From the managerial viewpoint of system, a two-tier queueing model is developed to investigate the dynamic performance of the supercomputer system under finite buffer control. The queueing system is formulated as a state dependent quasi-birth-and-death (QBD) process with two-dimensional state space. System performance measures can be obtained through the matrix geometric solution for such a QBD process.**

Keywords-finite buffer queue; matrix-geometric method; heterogeneous processors

I. INTRODUCTION

Supercomputers help researchers to discover real-world phenomenon through conducting large-scale scientific simulations except physical experiments. The implementations of heterogeneous system in a computer center has been lasting progress for past decades. The collection of multiprocessors is fed by a single common stream of batch jobs, where each job is dispatched to exactly one of the multiprocessor machines for processing. Examples of such distributed server systems include the Xolas distributed server at the MIT Lab for Computer Science, the Cray J90 distributed server at NASA Ames Research Lab, the Cray J90 distributed server at the Pittsburgh Supercomputing Center, and the Cray C90 distributed server at NASA Ames Research Lab (see Schroeder & Harchol-Balter [1] and reference therein).

Job scheduling on heterogeneous computer clusters is complicated, and how to utilize all heterogeneous server systems becomes an important research and managerial issue. In supercomputer centers today, designing a distributed server system (a collection of multiprocessors) often boils down to choosing the best task (computing jobs) assignment policy for the given supercomputer model and managerial requirements. As mentioned in Schroeder & Harchol-Balter [1], finding a good rule for assigning jobs to host machines remains an open question at many supercomputing sites. Over the last decades, many job assignment rules across a variety of distributed computational resources have been studied, e.g., Schaar & Efe [2], Chlamtac et al. [3], Feitelson et al. [4], von Laszewski [5], Piro et al. [6], Tang et al. [7], etc.

Various managerial goals for job assignment are usually conflicting, such as queueing efficiency versus system utilization. Tang et al. [7] studied scheduling policies for balancing workload. Downey [8] investigated the decision making on job assignment to a space-sharing parallel computer based on observed workloads. Under First-Come-First-Served scheduling policy, von Laszewski [5] demonstrated the advantage of well utilizing compute systems while submitting jobs across a variety of cluster. Bucur & Epema [9] conducted experimental analysis for the computational power of the existing systems when scheduling rigid jobs on a multicluster systems. Piro et al. [6] improved job scheduling strategies based on historical data of computing workloads to balance job assignment efficiently. But seldom works were evaluated analytically or systematically from the managerial viewpoint of queueing system under supercomputing workloads.

The main contribution of this paper is to provide an analytic mechanism for managing heterogeneous systems in a supercomputer center. We study a real supercomputing system Cray XE6m with two type of processors, which is currently running at the Supercomputing Research Center (SRC), National Cheng Kung University (NCKU), Taiwan. To manage computing resources properly, a queueing analysis of two-tier service system with finite buffer control is conducted in this paper. One type of processors offers guaranteed delay time with a finite buffer space, and the other type offers service with unlimited waiting space. We develop a two-dimensional state dependent quasi-birth-and-death (QBD) queueing model for evaluating the impact of the finite buffer control on the system performance measures, e.g., average queue length and average waiting time.

II. ARCHITECTURE OF CRAY XE6M

Since Year 2012, a supercomputer Cray XE6m in the NCKU-SRC has been offering computing services for conducting academic research. In Year 2011, the XE6m model

became commercially available, where "m" stands for midrange with at most 6 cabinets and interconnect network support with YARC switch [10]. The Cray XE6m in NCKU-SRC possesses two type heterogeneous processors (Abu Dhabi and Istanbul), and is divided into two subsystem managed with PBS job scheduler (as shown in Table 1). Besides, a two-dimensional torus architecture is employed in the system as the interconnection network. Cray Gemini network design reaches 160 GB/s. The main mission of NCKU-SRC is to provide broad computing services to research community, including large-scale scientific MPI applications in various research fields. Examples of large-scale scientific applications include the design of disaster prevention technology, numerical simulation of weather research and forecasting, biomedical research of molecular dynamics, next generation sequencing analysis, and first principle simulation, etc. For dealing with computer simulations in these research works, related simulation software packages have also been installed and executed at this supercomputer center NCKU-SRC. Main software packages include WRF, NAMD, LAMMPS, CPMD, SOAPdenovo, Bowtie, ABINIT, etc. Users execute qsub command with running script to submit computing jobs. The resource scheduling is managed by PBS Pro with two type of configuration.

TABLE I. SPECIFICATIONS OF TWO TYPE OF PROCESSORS IN CRAY XE6M.

Specifications	**Type-1 (Abu Dhabi)**	**Type-2 (Istanbul)**
Peak Performance (Rpeak)	2.45 Tflops	2.00 Tflops
Operating System	Cray Linux	Cray Linux
Number of Nodes	8 nodes	16 nodes
Memory per Node (GB)	64 GB	32 GB
Total Memory (GB)	512 GB	512 GB
Processor Clock Rate	2.4 GHz	2.6 GHz
Cores per Node	16 cores	6 cores
Total CPU Cores	128 cores	96 cores
Network Architecture	2 blades, YARC	4 blades, YARC
Storage System	Lustre / Fibre channel	Lustre / Fibre channel
Storage Capacity	20TB/RAID5	20TB/RAID5

III. A TWO-TIER SERVICE QUEUEING MODEL

In this section, the queueing effect of job assignment on two type of processors in Cray XE6m system will be evaluated systematically. We consider a two-tier service queueing model with two type of processors (Abu Dhabi and Istanbul) for job executions, as shown in Figure 1. The collection of multiprocessors is fed by a single common stream of computing jobs, where each job is dispatched to exactly one type of the multiprocessor machines for processing. Suppose the arrival of computing jobs at the login node follows a Poisson process with mean rate λ. Type-1 processors offer service without limited waiting space, and Type-2 processors offer guaranteed delay time with a finite waiting space, K.

Assume computing jobs are dispatched immediately upon arrival to one of these two type of processors, and jobs are run in First-Come-First-Served order. Meanwhile, for each type of processors, the whole processors can only be occupied by one job at a time. That is, jobs have exclusive access to host processors. Jobs are assumed to be run-to-completion (i.e., no preemption, no time-sharing). As mentioned in Schroeder & Harchol-Balter [1], run-to-completion is the common mode of operation in supercomputing environments. Without loss of generality, we assume the two service rates, denoted by μ_1 and μ_2 respectively, are different for these two type of processors and running time of jobs follows exponential distributions. Hence, this two-tier service queueing model can be formulated as a state dependent QBD process, where the arrival of computing jobs depends on queueing states of the system when it is not congested.

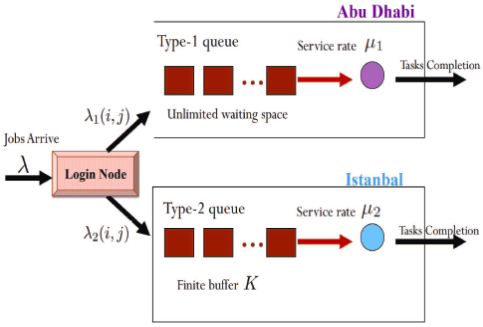

FIGURE I. A TWO-TIER SERVICE QUEUEING MODEL.

Let $q_1(t)$ and $q_2(t)$ be the queue lengths (including the job in service) of Type-1 processors and Type-2 processors, respectively. We define the system state as $(q_1(t), q_2(t))$ on the state space

$$\boldsymbol{\Omega} = \{(i,j) \mid i = 0,1,\dots;\ j = 0,1,\dots,K\}.$$

At system state (i,j), suppose that a job is assigned to Type-1 processors with a probability, which results in a state-dependent arrival rate $\lambda_1(i,j)$. Similarly, there is a state-dependent arrival rate $\lambda_2(i,j)$ at Type-2 processors. Note that $\lambda_1(i,j) + \lambda_2(i,j) = \lambda$ for all states (i,j). The states of the presented queueing system can be classified into three categories: the states in category I are "all join Type-1 processors" states; the states in category II are "join Type-1processors or Type-2 processors" states; and the states in category III are "all join Type-2 processors". The transition rate from state (i,j) to state (i',j') is given by the elements of the infinitesimal generator matrix Q. Based on the classification of the states, we can specify the matrix Q as follows:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_{0,0} & \mathbf{C}_{0,1} & & & & & & \\ \mathbf{A} & \mathbf{B}_{1,1} & \mathbf{C}_{1,2} & & & & & \\ & \mathbf{A} & \mathbf{B}_{2,2} & \mathbf{C}_{2,3} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{A} & \mathbf{B}_{K,K} & \mathbf{C}_{K,K+1} & & \\ & & & & \mathbf{A} & \mathbf{B} & \mathbf{C} & \\ & & & & & \ddots & \ddots & \ddots \end{bmatrix},$$

where all elements are $(K+1) \times (K+1)$ matrices. Detailed structure of sub-matrices in the infinitesimal generator matrix Q can be found in Luh et al. [11].

Under the stability condition, the stationary probability vector is defined as

$$\boldsymbol{\pi}_n = [\pi_{n,0}, \pi_{n,1}, \ldots, \pi_{n,K}],$$

where indices n = 0,1, ... , denote steady states of Type-1 queue. When n ≥ K, the matrix geometric solution for such a QBD process can be obtained by

$$\boldsymbol{\pi}_{n+1} = \boldsymbol{\pi}_n \mathbf{R}, \quad (1)$$

where R is the rate matrix. Like any regular QBD process, the rate matrix R should satisfy R2A + RB + C = 0 and can be solved by using one of many known algorithms. Interested readers may refer to Neuts [12] and references therein. For 0 ≤ n ≤ K, the probability vector π_n can be obtained by solving a set of equations. From πQ = 0 and (1), the steady-state vectors $\pi_0, \pi_1, \ldots$ and π_K can be solved from the boundary conditions (2)-(6) and the normalization condition (7) as follows:

$$\boldsymbol{\pi}_0 \mathbf{B}_{0,0} + \boldsymbol{\pi}_1 \mathbf{A} = 0, \quad (2)$$

$$\boldsymbol{\pi}_0 \mathbf{C}_{0,1} + \boldsymbol{\pi}_1 \mathbf{B}_{1,1} + \boldsymbol{\pi}_2 \mathbf{A} = 0, \quad (3)$$

$$\boldsymbol{\pi}_1 \mathbf{C}_{1,2} + \boldsymbol{\pi}_2 \mathbf{B}_{2,2} + \boldsymbol{\pi}_3 \mathbf{A} = 0, \quad (4)$$

$$\vdots$$

$$\boldsymbol{\pi}_{K-2} \mathbf{C}_{K-2,K-1} + \boldsymbol{\pi}_{K-1} \mathbf{B}_{K-1,K-1} + \boldsymbol{\pi}_K \mathbf{A} \quad (5)$$

$$\boldsymbol{\pi}_{K-1} \mathbf{C}_{K-1,K} + \boldsymbol{\pi}_K (\mathbf{B}_{K,K} + \mathbf{RA}) = 0, \quad (6)$$

$$\boldsymbol{\pi}_0 \mathbf{1} + \boldsymbol{\pi}_1 \mathbf{1} + \cdots + \boldsymbol{\pi}_K (\mathbf{I} - \mathbf{R})^{-1} \mathbf{1} = 1, \quad (7)$$

After determining the stationary distribution, we can obtain the major system performance measures, including average queue length and average waiting time. In the case of large-scale supercomputing system with huge buffer size K, it would result in a large number of boundary states and a large number of phases of the QBD process. It would greatly increase the computational complexity and may cause the ill-conditioned matrices of the traditional iterative algorithm for the rate matrix. To overcome this challenge, an efficient algorithm proposed by Luh et al. [11] could be applied directly to the case of medium to large scale buffer size K for determining the stationary distribution in large-scale supercomputing systems.

IV. Numerical Experiments

We consider two scenarios in this numerical experiments, that is, heavy traffic load and light traffic load. We demonstrate the effect of finite buffer control on the system performances, i.e., average queue length and average waiting time. For the heavy traffic load, we have $\lambda = 1$, $\mu_1 = 0.6$ and $\mu_2 = 0.8$; on the other hand, we take $\lambda = 0.8$ for the light traffic load. The numerical experiments are conducted through computing language MATLAB on the PC platform with Intel® Core™ i7-3770 CPU @ 3.40 GHz and 32 GB RAM.

As the finite buffer size K varies from 2 to 15, Figure 2 shows the average queue length of jobs for Type-1 processors and Type-2 processors, individually. The effect of increasing the buffer size on deducing average waiting time is illustrated in Figure 3. As the finite buffer size is large enough, it can be found that the benefit of increasing the finite buffer on improving system performances would decrease.

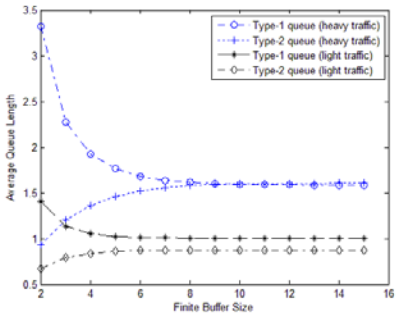

FIGURE II. AVERAGE QUEUE LENGTH VERSUS THE BUFFER SIZE.

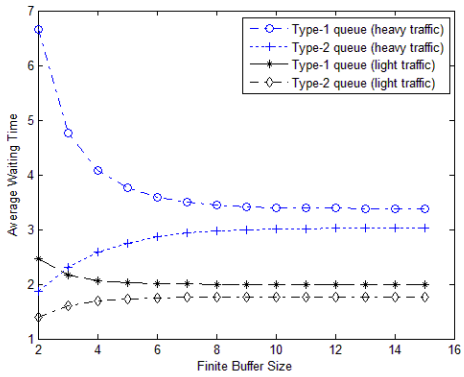

FIGURE III. AVERAGE WAITING TIME VERSUS THE BUFFER SIZE.

V. Conclusions

We present a queueing model based on a real supercomputer system and observed workloads at NCKU-SRC. The state dependent QBD process derived in this paper allows us to investigate the dynamic performance of such a two-tier service supercomputer system. We draw a conclusion that proper buffer control can balance queuing efficiency and the cost of enlarging the finite buffer size. According to the presented queueing analysis, managers could assign arriving jobs to appropriate type of processors in order to achieve the better system performance. The queueing analysis conducted in this study could help in developing a management scheme for job assignment on heterogeneous supercomputer systems with finite buffer control.

Acknowledgements

The authors would like to thank Prof. Yuefan Deng and Dr. Sing-Wu Liou for useful discussions on this work. This research was partially supported by the National Science Council, Taiwan, R.O.C., under grant number NSC 102-2911-I-006-301.

References

[1] Schroeder, B. & Harchol-Balter, M., Evaluation of task assignment policies for supercomputing servers: The case for load unbalancing and fairness, *Cluster Computing*, vol. 7, pp. 151-161, 2004.

[2] Schaar, M.A. & Efe, K., Effective queueing strategies for co-scheduling in a pool of processors, *Computer Communications*, vol. 19, pp. 743-753, 1996.

[3] Chlamtac, I. Kienzle, M.G. & Szabo, C., Characterizing the behaviour of high-speed interconnection systems with distributed control, *Telecommunication Systems*, vol. 6, pp. 91-115, 1996.

[4] Feitelson, D.G., Rudolph, L., Schwiegelshohn, U., Sevcik, K.C. & Wong, P., Theory and practice in parallel job scheduling, in *Proceedings of the 3rd Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 1-34, 1997.

[5] von Laszewski, G., A loosely coupled metacomputer: co-operating job submissions across multiple supercomputing sites, *Concurrency Practice and Experience*, vol. 11, pp. 933-948, 1999.

[6] Piro, R.M., Guarise, A., Patania, G. & Werbrouck, A., Using historical accounting information to predict the resource usage of grid jobs, *Future Generation Computer Systems*, vol. 25, pp. 499-510, 2009.

[7] Tang, W., Ren, D., Lan, Z. & Desai, N., Toward balanced and sustainable job scheduling for production supercomputers, *Parallel Computing*, vol. 39, pp. 753-768, 2013.

[8] Downey, A.B., Using queue time predictions for processor allocation, in *Proceedings of the 3rd Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 35-57, 1997.

[9] Bucur, A.I. & Epema, D.H., The influence of the structure and sizes of jobs on the performance of co-allocation, *Lecture Notes in Computer Science*, vol. 1911, pp. 154-173, 2000.

[10] Kerbyson, D.J., Barker, K.J., Vishnu, A. & Hoisie, A., A performance comparison of current HPC systems: Blue Gene/Q, Cray XE6 and InfiniBand systems, *Future Generation Computer Systems*, vol. 30, pp. 291-304, 2014.

[11] Luh, H., Zhang, Z. & Wang, C.H., A computing approach to two competing services with a finite buffer effect, in *Proceedings of the 8th International Conference on Queueing Theory and Network Applications*, pp. 15-21, 2013.

[12] Neuts, M.F., *Matrix-Geometric Solutions in Stochastic Models*. The John Hopkins University Press, 1981.