DOI:10.5392/IJoC.2010.6.4.030

Performance Evaluation of Gang Scheduling Policies with Migration in a Grid System

Cheul Woo Ro, Yang Cao
Dep. of Computer Eng.
Silla University, Busan, KOREA

ABSTRACT

Effective job scheduling scheme is a crucial part of complex heterogeneous distributed systems. Gang scheduling is a scheduling algorithm for grid systems that schedules related grid jobs to run simultaneously on servers in different local sites. In this paper, we address grid jobs (gangs) schedule modeling using Stochastic reward nets (SRNs), which is concerned for static and dynamic scheduling policies. SRN is an extension of Stochastic Petri Net (SPN) and provides compact modeling facilities for system analysis. Threshold queue is adopted to smooth the variations of performance measures. System throughput and response time are compared and analyzed by giving reward measures in SRNs.

Keywords*:* gang Scheduling, grid System, stochastic reward nets

1. INTRODUCTION

Grid environment enables various applications to share loosely coupled resources and services scattered everywhere. Grid job scheduling is the key technology in grid resource allocation. A grid job (gang) consists of a number of parallel tasks that must be served concurrently [1]. How to effectively schedule gangs to available grid resources is a crucial problem for a grid system because of its dynamic, heterogeneous and autonomous features [2]. The design of gang scheduling scheme that can adjust the behavior and response of a system to meet certain performance requirements is thus a challenging problem.

Threshold-based queue with hysteresis has many applications in the dynamic control of computer systems and communication networks [3]. This kind of queue is controlled by a sequence of forward thresholds and a sequence of reverse thresholds. Thresholds help to make the variations of delay and throughput smoothly, and hysteresis ensures that the control mechanism will not switch too much [4].

The Stochastic Petri Net Package (SPNP) [5] is a useful modeling tool for solution of SPN models. The SPN models specified to SPNP are actually Stochastic Reward Nets (SRNs) which are based on the “Markov Reward Models” paradigm. SRN has the ability to allow extensive marking dependency. It also has one important feature of expressing complex enabling/disabling conditions through guard functions [6]. Appropriate reward rates associated with the markings are assigned to SRN in order to get the performance analysis measures of a system.

In order to meet the need of grid environment to achieve effectiveness, in this paper we address three different gang scheduling policies, which are static Load Sharing (LS) and two dynamic Load Comparing (LC) schemes. Threshold-based queue with hysteresis is adopted to help the control of gang scheduler. We develop SRN model to capture gang’s behavior, compare and analyze these three policies by giving reward measures.

2. SRN AND THRESHOLD QUEUES

2.1 SRN

Stochastic Reward Net (SRN) is based on the Markov Reward Model (MRM) [7] which provides a powerful modeling environment for dependability analysis, performance analysis and performability modeling. SRN is an extension of Stochastic Petri net (SPN) [8,9] augmented with the ability to specify output measures as reward-based functions, for complex systems performance evaluation. SRN has the ability to allow extensive marking dependency. Through guard functions it can also express complex enabling/disabling conditions. This can simply give graphical representations to complex systems.

For an SRN, according to the expected values of the reward rate functions, all the output measures can be expressed. In order to analyze or evaluate the performance and reliability/availability of a system, we need to assign appropriate reward rates associated with markings to its SRN. When SRN is automatically converted into a MRM, the required measures of the original SRN can be made through

[*] *Corresponding author. E-mail : cwro@silla.ac.kr*

Manuscript received Oct. 21, 2010 ; accepted Dec. 13, 2010

steady state and/or transient analysis of the MRM [10].

2.2 Threshold Queue

Threshold queues have many applications in communication networks. Threshold-based service policies have been applied and proven to be optimal to queueing systems to control service rate and the number of servers taking a single queue as policies [11]. The queue is controlled by a sequence of forward thresholds and a sequence of reverse thresholds.

A K-server hysteresis threshold-based queueing system is considered that the number of servers is governed by a forward threshold vector $\boldsymbol{F}=(\boldsymbol{F}_1, \boldsymbol{F}_2,\ldots,\boldsymbol{F}_{k-1})$ and a reverse threshold vector $\boldsymbol{R}=(\boldsymbol{R}_1, \boldsymbol{R}_2,\ldots\boldsymbol{R}_{k-1})$. Normally following conditions should be achieved: $\boldsymbol{F}_1<\boldsymbol{F}_2<\ldots<\boldsymbol{F}_{k-1}$ and $\boldsymbol{R}_1<\boldsymbol{R}_2<\ldots<\boldsymbol{R}_{k-1}$ [12]. When an application arrives to the empty queue of grid scheduler, it is serviced by a single server. Whenever the number of applications exceeds a forward threshold $\boldsymbol{F}_i$, a server is added to the system and server activation is instantaneous. Whenever the number of applications falls below a reverse threshold $\boldsymbol{R}_i$, a server is removed from the system. So it is good for the system to use an “appropriate” number of servers so as to satisfy some performance requirements, such as the mean system response time.

When a system is moving towards a heavily loaded operation period, it is desirable to add servers, while it is desirable to remove servers when a system is moving towards a lightly loaded operation period [12].

3. SYSTEM MODELING

3.1 Model Description

The SRN models (grid system) in this paper aim to study the performance of 4 different grid scheduling policies. The system consists of one grid scheduler with its own waiting queue and two homogeneous local sites each of which has 16 servers. We assume that the two sites are connected through a wide area network, while the servers in each site are interconnected through a high speed local area network [1]. There are two levels of job scheduling in this system: grid and local. We assume that a gird job (gang) consists of a number of parallel tasks, while the local job has only one task with higher priority.

The structure of the SRN model is shown in Figure 1. The firing of transition ***T*** means gangs arriving with rate#λ, then grid scheduler (GS) stores incoming gangs temporarily in its own queue (place ***GQ***) waiting for scheduling. According to the scheduling policy, GS allocates gangs to different local sites. Transition $\boldsymbol{GS}_1$ or $\boldsymbol{GS}_2$ fires mean that tasks set of a gang is allocated to ***Site1*** or ***Site2***. Each site also has its own local job queue (place $\boldsymbol{LQ}_1$ and $\boldsymbol{LQ}_2$). Transition $\boldsymbol{LT}_1$ or $\boldsymbol{LT}_2$ fires mean that local jobs come and move to place $\boldsymbol{pl}_1$ or $\boldsymbol{pl}_2$. Since they are higher priority tasks, through immediate transition $\boldsymbol{it}_1$ or $\boldsymbol{it}_2$, they can immediately move to LQ1 to be executed by servers. Transition $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ fires mean that tasks (either local job’s task or gang’s tasks) are getting executed. Place $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ imply the capacity of each site. Transition $\boldsymbol{GS}_{11}$ and $\boldsymbol{GS}_{21}$ are migrations mechanism we adopt to transfer a job from one local queue to another queue to ensure all tasks of a gang will get execution simultaneously. This method will be explained in Section 3.2.

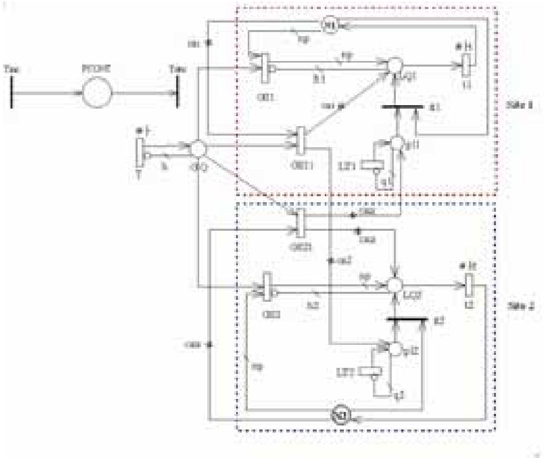

Fig. 1. SRN model of the Grid System and threshold queue with hysteresis.

3.2 Migration

Gang consists of a number of parallel tasks that must be executed simultaneously. Thus the delay of gang execution is due to that one or more of its tasks are waiting in queues which belong to busy or reserved servers [13]. Migration is a way to solve/avoid this kind of problem [14]. It implies in our case transferring of a task from one site to another site [15].

In our SRN model, migration is implemented by using ***guardfun***() to represent the enable condition of $\boldsymbol{GS_{11}}$ and $\boldsymbol{GS_{21}}$. We also use zigzag sign in accordance with ***guardfun***(). The zigzag sign $\boldsymbol{ca}_1$ on the input arc from $\boldsymbol{N}_1$ to $\boldsymbol{GS_{11}}$ denotes that the multiplicity of the arc is variable. Assume a gang contains 5 tasks (***np***=5) arrives, at that time idle servers in ***site1*** are $\boldsymbol{ca_1}$ (current number of tokens in $\boldsymbol{N}_1$, $0< \boldsymbol{ca_1}<\boldsymbol{np}$, which means transition $\boldsymbol{GS_1}$ can’t be fired). Through ***guardfun***(), $\boldsymbol{GS_{11}}$ gets fired by moving $\boldsymbol{ca_1}$ tokens from $\boldsymbol{N}_1$ to $\boldsymbol{LQ_1}$ which means $\boldsymbol{ca_1}$ tasks to be executed in ***site1***, and by moving $\boldsymbol{ca_2}$ tokens to place $\boldsymbol{pl}_2$ which means the other tasks are to be executed in ***site2*** (through firing transition $\boldsymbol{it}_2$, they are moved to $\boldsymbol{LQ}_2$ to be executed immediately), and vice versa. Transition $\boldsymbol{LT_2}$ fires means that local job arrives at the local scheduler. Because local job has higher priority, it can soon get executed. The ***guardfun()*** we set for $\boldsymbol{GS_{11}}$ and the variable of zigzag sign are shown in Table 1 and 2 respectively.

Table 1. Guard functions of $\boldsymbol{GS_{11}}$ and $\boldsymbol{GS_{21}}$.

Transition	guard
$\boldsymbol{GS_{11}}$	$(1< \#\boldsymbol{N_1} < \boldsymbol{np})$ && $(\#\boldsymbol{N_2} > (\boldsymbol{np} - \#\boldsymbol{N_1}))$
$\boldsymbol{GS_{21}}$	$(1< \#\boldsymbol{N_2} < \boldsymbol{np})$ && $(\#\boldsymbol{N_1} > (\boldsymbol{np} - \#\boldsymbol{N}_2))$

Table 2. Zigzag Sign Variables.

Zigzag Sign Variable	Value
$\boldsymbol{ca_1}$	$\#\boldsymbol{N_1}$
$\boldsymbol{Ca_2}$	$\boldsymbol{np} - \#\boldsymbol{N_1}$
$\boldsymbol{ca_{11}}$	$\#\boldsymbol{N_2}$
$\boldsymbol{Ca_{21}}$	$\boldsymbol{np} - \#\boldsymbol{N_2}$

3.3 Threshold Queue with Hysteresis

The use of queue thresholds is developed for network traffic congestion control [16]. An typical example of threshold queue with hysteresis is the ***K***-server queue in which additional servers arrive when the buffer content exceeds the forward thresholds and leave when the buffer content decrease to reverse thresholds.

According to Tuffin's model [4], we consider in this paper a Markovian threshold queue with hysteresis and capacity ***C*** controlled by a set of forward thresholds ($\boldsymbol{S_1}$, $\boldsymbol{S_2}$, … $\boldsymbol{S_k}$) and a set of reverse thresholds ($\boldsymbol{s_1}$, $\boldsymbol{s_2}$,…s_k) in order to control the throughput and delay. Figure 1 shows the SPN model related to this single queue. Place ***PCONT*** gives the rate control state changing through the immediate transitions when thresholds are reached (events given by the guard functions of transition ***T***inc and ***T***dec). Table 3 shows the guard functions and Table 4 shows the rate functions.

Table 3. Guard functions of the SRN of Figure 2.

Transition	guard
Tinc	$\#GQ>\boldsymbol{S}_{\#\mathrm{PCONT}}\&\&\#PCONT<K$
Tdec	$\#GQ<=\boldsymbol{s}_{\#\mathrm{PCONT-1}}$
T	$\#GQ<C$

Table 4. Rate functions of the SRN of Figure 2.

Transition	Rate
T	$\lambda*(\#PCONT)$
GS1	$\mu 1*(\#PCONT)$
GS2	$\mu 2*(\#PCONT)$

4. SCHEDULING POLICIES

4.1 Load Sharing (LS)

Load sharing is a scheduling policy to assign to each server evenly work proportional to its performance, thereby minimizing the response time of a job, enhancing resource utilization and improving throughput [17]. The main goal of load sharing is to provide a distributed, low cost scheme that balances the load across all the servers. For the static LS scheduling policy used in this paper, grid jobs are scheduled evenly between ***Site****1* and ***Site****2* through firing transition $\boldsymbol{GS_1}$ or $\boldsymbol{GS_2}$ in turn.

4.2 Load Comparing-1 (LC-1)

Due to the dynamic nature of Grids and the lack of information on resources and jobs, more flexible scheduling policies should be created to achieve higher system performance. In this paper, we present two dynamic scheduling policies.

For the dynamic LC-1 scheduling policy, we compare the current loads of ***Site****1* and ***Site****2*. Jobs are scheduled to the site with smaller load to accelerate system running. This is done by using ***guardfun***() to represent the enable condition of transitions $\boldsymbol{GS_1}$ and $\boldsymbol{GS_2}$.

4.3 Load Comparing-2 (LC-2)

For the dynamic LC-2 scheduling policy, we set two thresholds $\boldsymbol{th_1}$ and $\boldsymbol{th_2}$, and compare the Total Loads (TL) of ***Site****1* and ***Site****2*. If ***TL***<= $\boldsymbol{th_1}$, jobs will be scheduled to ***Site****1*, else jobs will be scheduled to ***Site****2*. This is done by using ***guardfun***() to represent the enable condition of transitions $\boldsymbol{GS_1}$ and $\boldsymbol{GS_2}$.

The ***guardfun()*** we set for $\boldsymbol{GS_1}$ under two dynamic policies are shown in Table 5.

Table 5. Guard functions of transition $\boldsymbol{GS_1}$.

Dynamic Policy	Transition	guard
LC-1	$\boldsymbol{GS_1}$	$\#\boldsymbol{LQ}1\leq\#\boldsymbol{LQ}2$
	$\boldsymbol{GS_2}$	$\#\boldsymbol{LQ}1>\#\boldsymbol{LQ}2$
LC-2	$\boldsymbol{GS_1}$	$(\#\boldsymbol{LQ1}+\#\boldsymbol{LQ2})\leq\boldsymbol{th1}$
	$\boldsymbol{GS_2}$	$\boldsymbol{th1}<(\#\boldsymbol{LQ1}+\#\boldsymbol{LQ2})\leq\boldsymbol{th2}$

5. PERFORMANCE COMPARISON

5.1 Measures of Interest

In order to evaluate and compare the system's performance under different scheduling policies from different points of view, following common measures will be employed. The measures are defined in terms of reward rates associated with the markings of the SRN. The SRN model based on MRM paradigm is specified to the software package SPNP so that we can obtain the interested numerical results [5]. We assume that all transition firing rates in our SRN models are exponentially distributed, we perform the steady-state analysis of the model we have constructed.

- System Throughput (ST)

ST is the total number of tasks to be processed. To calculate the total system throughput, we use the following formula, where ***rate()*** is the SRN function representing the actual firing rate of the transition.

$$ST=\sum_{i=1}^{2} rate(t_i) \tag{1}$$

- System Response Time (SRT)

SRT is the time the system takes to fulfill whole tasks. To calculate the total system response time, we use Little's formula. The ***rate("T")*** is the actual arriving rate of gangs. The ***mark()*** is the SRN function representing the number of tokens in system queue, including grid queue and local queue.

$$SRT=\frac{\sum_{i=1}^{2} mark(LQ_i)+mark(GQ)}{rate("T")} \tag{2}$$

- Utilization of Servers (UoS)

We define a formula to represent servers' utilization of these two sites. The formula is as the following where $\boldsymbol{\#N_i}$ (number of tokens in N_1) represents the idle servers in each site, $\boldsymbol{N_1}+\boldsymbol{N_2}$ represents the total number of servers in both sites.

$$UoS = 1 - \frac{\sum_{i=1}^{2} \# N_i}{N1 + N2} \qquad (3)$$

- Grade of Service (GoS)

We then define a formula to represent grade of service. ***W*** is the weight to take use of SRT. In our case, ***W***=10. The formula is as the following:

$$GoS = ST + \frac{W}{SRT} \qquad (4)$$

5.2 Numerical Results

A gang consists of a number of parallel tasks. In this study, the number of tasks a gang can have is limited to 5 in order to decrease model running time. According to Tuffin [4] we give the gang queue maximum capacity C=64, the values of forward thresholds ($\boldsymbol{S_1}$, $\boldsymbol{S_2}$, … $\boldsymbol{S_{k-1}}$) and reverse thresholds ($\boldsymbol{s_1}$, $\boldsymbol{s_2}$,…$s_{\boldsymbol{k-1}}$) are (0,8,16,24,32,44) and (0,4,8,12,16,20) respectively (k=7). We assume a total of 32 servers exits in the system (each sites consist of S=16 servers). We also give some different input parameters ***f***(λ, *n1, n2, q1, q2, h1, h2, np, th1, th2*). The corresponding values are: $\lambda \in$ {1.0,1.5…,5.5}; *n1=n2=16, q1=q2=2, np=5, h1=n1+q1, h2=n2+q2, th1=(h1+h2)/3, th2=(h1+h2)*2/3.*

Figure 2-5 show ST, SRT, UoS and GoS respectively under three scheduling policies with different gangs arriving rate λ.

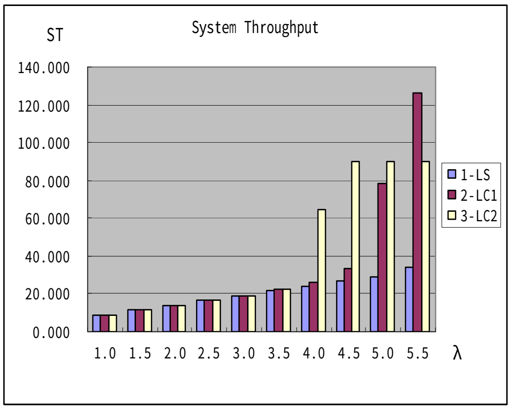

Fig. 2. ST comparison.

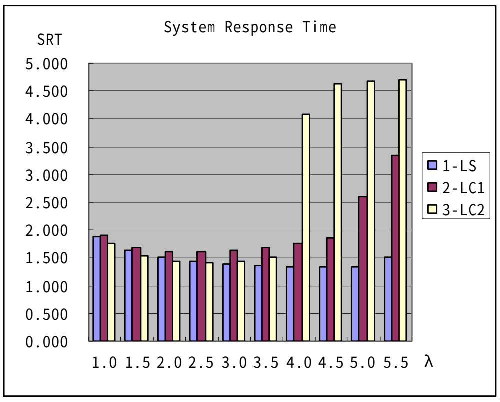

Fig. 3. SRT comparison.

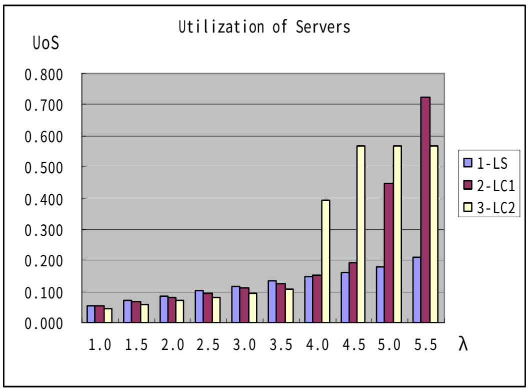

Fig. 4. UoS comparison.

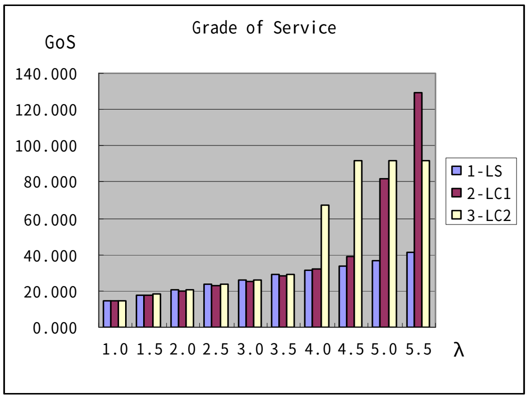

Fig. 5. GoS comparison.

As we expected, with the gangs arriving rate λ increasing, the ST of dynamic scheduling such as LC-1 or LC-2 gets better than that in static scheduling such as LS. For the SRT, when λ <3.0, LC-2 gets better than that of LS and LC-1, but with λ increasing and throughput changing larger, LS gets better than these two dynamic policies.

6. CONCLUSION

In this paper, three different gang scheduling policies have been applied in a grid system model. Threshold queue is adopted to smooth the variations of delay and throughput. Migrations are implemented to avoid delay of gang execution. We compare these policies and analyze their performances such as system throughput and system response time by giving reward measures in SRN. The results show that two dynamic scheduling policies, especially the second one, present better performance than the static Load Sharing scheduling policy, thus they fit scheduling jobs in a heterogeneous grid computing system well.

There are still lots of work to do with that system, such as create more excellent scheduling policies to achieve higher system performance and how to define thresholds to get system optimization. We will tackle on these situations in our future work and get more deep and complicated analysis.

REFERENCES

[1] Sofia K. Dimitriadou, Helen D. Karatza, "Multi-Site Allocation Policies on a Grid and Local Level", Electronic Notes in Theoretical Computer Science. Volume 261:163-179, 2010.

[2] Peijie Huang, Hong Peng, Piyuan Lin, Xuezhen Li, "Static strategy and dynamic adjustment: An effective method for grid task scheduling", Future Generation Computer Systems 25, 2009, pp. 884-892.

[3] L.M. Le Ny and B. Tuffin, "Modeling and analysis of multi-class threshold-based queues with hysteresis using Stochastic Petri Nets", In Proceedings the International Conference on Applications and Theory of Petri Nets. Lecture Notes in Computer Science, Springer Verlag, 2002.

[4] B. Tuffin and L.M. Le Ny, "Modeling and analysis of threshold queues with hysteresis using stochastic Petri nets: the monoclass case", In Proceedings of Petri Nets and Performance Models, PNPM'01, pages 175-184, IEEE CS Press, Aachen, Germany, 2001.

[5] C. Hirel, B. Tuffin, and K. S. Trivedi, "SPNP: Stochastic Petri Nets. Version 6.0", in Computer performance evaluation: Modelling tools and techniques; 11th International Conference; TOOLS 2000, Schaumburg, Il., USA, B. Haverkort, H. Bohnenkamp, C. Smith(eds.), Lecture Notes in Computer Science 1786, Springer Verlag, 2000.

[6] Malhotra, M. and Ciardo, G, "Dependability Modeling Using Petri-Net", IEEE Transactions on Reliability, 44(3):428-440,1995.

[7] Performance analysis of the CORBA Event Service using stochastic reward nets, S. Ramani, K. S. Trivedi, B. Dasarathy, Proc. of the 19th IEEE Symposium on Reliable Distributed Systems, pp 238-247, 2000.

[8] Peter J. Haas, "Stochastic Petri Nets for modelling and SRN", Winter SRN Conference Proceedings of the 36th conference on Winter SRN, SESSION: Advanced tutorials: Stochastic Petri Nets. pp.101-112, 2004.

[9] Stochastic Reward Nets for Reliability Prediction, Jogesh Muppala, Gianfranco Ciardo, and K. S. Trivedi, Communications in Reliability, Maintainability and Serviceability: An International Journal published by SAE International, Vol. 1, No. 2, pp. 9-20, July 1994.

[10] Cheul Woo Ro, Kyung Min Kim, "Stochastic Petri Nets Modeling Methods of Channel Allocation in Wireless Networks," IJOC(International Journal of Contents) Vol. 4, No.3, 2008.9.

[11] O. C. Ibe, J. Keilson, "Multi-server threshold queues with hysteresis", Performance Evaluation 21:185-213, 1995.

[12] J.C.S.Lui, L. Golubchik, "Stochastic complement analysis of multi-server threshold queues with hysteresis, Performance Evaluation", 35:185-213, 1999.

[13] Margo, M.W., Yoshimoto, K., Kovatch, P., Andrews, P., "Impact of reservations on production job scheduling. In: Job Scheduling Strategies for Parallel Processing". Springer, Berlin, Heidelberg, pp. 116-131, 2008.

[14] Wang, X.Y., Zhu, Z.Y., Du, Z.H., Li, S.L., "Multi-cluster load balancing based on process migration", Lecture Notes in Computer Science, vol. 4847. Springer-Verlag, Berlin, Heidelberg, pp. 100-110, 2007.

[15] Milojičić,D.S., Douglis, F., Paindaveine, Y., Wheeler, R., Zhou, S., "Process Migration", ACM Computing Surveys (CSUR), vol. 32 (3). ACM, New York, pp. 241-299, 2000.

[16] Irfan Awan, "Analysis of multiple-threshold queues for congestion control of heterogeneous traffic streams", Simulation Modelling Practice and Theory. 14:712-724, 2006.

[17] Saravanakumar E. and Gomathy Prathima, "A novel load balancing algorithm for computational grid", International Journal of Computational Intelligence. Volume 1, Issue 1, pp. 20-26, 2010.

Cheul Woo Ro

Refer to IJOC, Vol.4, No. 3, 2008.9

Yang Cao

She received the M.S in Belgium in 2003. She works at Eastern Liaoning University in China. She is currently a PhD candidate in the department of computer engineering at Silla University. Her main research interests include modeling and analysis of communication systems.