Supplementary Material for Optimal Multiserver Configuration for Profit Maximization in Cloud Computing

Junwei Cao, *Senior Member, IEEE* Kai Hwang, *Fellow, IEEE* Keqin Li, *Senior Member, IEEE*
Albert Y. Zomaya, *Fellow, IEEE*

1 Simulation Settings and Results

Simulations have been conducted for two purposes, namely, (1) to validate our analytical results (Theorems 1 and 2); (2) and to find more effective queueing disciplines which increase the net profit of a service provider.

In Table 1, we show our simulation results by using the same parameters in Figures 1–10. For each $\lambda = 6.05, 6.15, ..., 6.95$, we trace the behavior of an M/M/m queueing system with the FCFS queueing discipline by generating a Poisson stream of service requests with arrival rate λ, recording the waiting and response times of each service request, and calculating the service charge to each service request. The average service charge of 1,000,000 service requests is reported in Table 1 for each λ. Notice that the maximum 99% confidence interval of all the data in the table is $\pm$0.5165372%. The analytical data in the table are obtained by Theorem 2 to calculate the expected charge to a service request. It is easily observed that our simulation results match with the analytical data very well. These results validate our theoretically predicted service charge in Theorem 2, which is based on our analytical result on waiting time distribution in Theorem 1.

Our analysis in this paper is based on the FCFS queueing discipline. A different queueing discipline may change the distribution of the waiting times, and thus, changes the average task response time and the expected service charge. Since the cost of a service provider remains the same, an increased/decreased expected service charge to a service request increases/decreases the expected net business gain of a service provider. To show the effect of queueing disciplines on the net profit of a service provider, we only need to show the effect of queueing disciplines on the expected service charge to a service request. We consider two simple queueing disciplines, namely,

- Shortest Task First (STF): Tasks (service requests) are arranged in a waiting queue in the increasing order of their task execution requirements;
- Largest Task First (LTF): Tasks (service requests) are arranged in a waiting queue in the decreasing order of their task execution requirements.

While other queueing disciplines can also be considered, these two disciplines are already very encouraging.

In Table 1, we also display our simulation results for STF and LTF by using the same parameters for FCFS. For each $\lambda = 6.05, 6.15, ..., 6.95$, we trace the behavior of an M/M/m queueing system with the STF and the LTF queueing disciplines respectively. The average service charge of 1,000,000 service requests is reported in Table 1 for each λ. We have the following observations.

- STF performs consistently better than FCFS. Furthermore, while the expected service charge drops significantly for FCFS when λ is close to the saturation point and the average waiting time becomes very long, the expected service charge of STF is still close to $a\bar{r}$ when λ is large.
- LTF performs worse than FCFS when λ is not very large. However, when λ is close to the saturation point, LTF performs better than FCFS in the sense that the expected service charge of LTF does not drop significantly when λ is large.

- *J. Cao is with the Research Institute of Information Technology, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: jcao@tsinghua.edu.cn*
- *K. Hwang is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA. E-mail: kaihwang@usc.edu*
- *K. Li is with the Department of Computer Science, State University of New York, New Paltz, New York 12561, USA. E-mail: lik@newpaltz.edu*

This is the author for correspondence.

- *A. Y. Zomaya is with the School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia. E-mail: albert.zomaya@sydney.edu.au*

Table 1: Simulation Results on the Expected Service Charge.

λ	Analytical	FCFS	STF	LTF
6.05	9.8245499	9.8185300	9.9792709	9.5841297
6.15	9.7798220	9.7762701	9.9651989	9.5239829
6.25	9.7193304	9.7151778	9.9565597	9.4591742
6.35	9.6351404	9.6384268	9.9514632	9.3984495
6.45	9.5136508	9.5015511	9.9015669	9.3375343
6.55	9.3298719	9.3432460	9.8674891	9.2532065
6.65	9.0334251	9.0317035	9.8038810	9.1751988
6.75	8.5084481	8.4933564	9.7186839	9.0780185
6.85	7.4277447	7.4383065	9.5750764	8.9842690
6.95	4.4400583	4.4744746	9.3974538	8.8743559

Unfortunately, due to lack of an analytical result on waiting time distribution similar to Theorem 1 for STF and LTF, the analytical work conducted in this paper for FCFS cannot be duplicated for STF and LTF. This can be an interesting subject for further investigation.

2 Proofs of Theorems 1 and 2

Proof of Theorem 1. If there are $k < m$ tasks in the queueing system when a new service request arrives, the waiting time of the service request is $W_k = 0$. The pdf of W_k can be represented as

$$f_{W_k}(t) = u(t),$$

for all $0 \le k \le m-1$. Furthermore, we have

$$\overline{W}_k = \lim_{z\to\infty} \frac{1}{2z} = 0,$$

for all $0 \le k \le m-1$.

If there are $k \ge m$ tasks in the queueing system when a new service request arrives, then the service request must wait until a server is available. Notice that due to the memoryless property of an exponential distribution, the remaining execution time of a task is always the same random variable as before, i.e., the original task execution time x with pdf $f_x(t) = \mu e^{-\mu t}$, no matter how long the task has been executed. Let $x_1, x_2, ..., x_m$ be the remaining execution times of the m tasks in execution when a new service request arrives. Then, we have $f_{x_j}(t) = \mu e^{-\mu t}$, for all $1 \le j \le m$.

It is clear that $y = \min\{x_1, x_2, ..., x_m\}$ is the time until the next completion of a task. Since

$$\boldsymbol{P}[y \ge t] = \prod_{j=1}^{m} \boldsymbol{P}[x_j \ge t] = \prod_{j=1}^{m} e^{-\mu t} = e^{-m\mu t},$$

we get

$$F_y(t) = \boldsymbol{P}[y \le t] = 1 - \boldsymbol{P}[y \ge t] = 1 - e^{-m\mu t},$$

and $f_y(t) = m\mu e^{-m\mu t}$, that is, y is also an exponential random variable with mean $1/m\mu$. The time until the next completion of a task is always the same random variable y, i.e., the minimum value of m i.i.d. exponential random variables with pdf $f_y(t) = m\mu e^{-m\mu t}$.

Notice that due to multiple servers, a task does not need to wait until all tasks in front of it are completed. Actually, the waiting time W_k of a task (under the condition that there are $k \ge m$ tasks in the queueing system when the task arrives) is $W_k = y_1 + y_2 + \cdots + y_{k-m+1}$, where $y_1, y_2, ..., y_{k-m+1}$ are i.i.d. exponential random variables with the same pdf $f_y(t) = m\mu e^{-m\mu t}$. The reason is that after $k-m+1$ completions of task executions, a task is at the front of the waiting queue and there is an available server, and the task will be scheduled to be executed. It is well known that $y_1 + y_2 + \cdots + y_k$ has an Erlang distribution whose pdf is

$$\frac{m\mu(m\mu t)^{k-1}}{(k-1)!} e^{-m\mu t}.$$

Hence, we get the pdf of W_k

$$f_{W_k}(t) = \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t},$$

for all $k \ge m$. Notice that $\bar{y} = 1/m\mu$ and

$$\overline{W}_k = (k-m+1)\bar{y} = \frac{k-m+1}{m\mu} = (k-m+1)\frac{\bar{x}}{m},$$

for all $k \ge m$.

Summarizing the above discussion, we obtain the pdf of the waiting time W of a service request as follows:

$$\begin{aligned}
&f_W(t) \\
&= \sum_{k=0}^{\infty} p_k f_{W_k}(t) \\
&= \left(\sum_{k=0}^{m-1} p_k\right) u(t) + \sum_{k=m}^{\infty} p_k \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t} \\
&= (1-P_q)u(t) + \sum_{k=m}^{\infty} p_0 \frac{m^m \rho^k}{m!} \cdot \frac{m\mu(m\mu t)^{k-m}}{(k-m)!} e^{-m\mu t} \\
&= (1-P_q)u(t) \\
&\qquad + p_0 \frac{m^m \rho^m}{m!} m\mu e^{-m\mu t} \sum_{k=m}^{\infty} \frac{\rho^{k-m}(m\mu t)^{k-m}}{(k-m)!} \\
&= (1-P_q)u(t) + p_0 \frac{(m\rho)^m}{m!} m\mu e^{-m\mu t} \sum_{k=0}^{\infty} \frac{(\rho m\mu t)^k}{k!} \\
&= (1-P_q)u(t) + p_m m\mu e^{-m\mu t} e^{\rho m\mu t} \\
&= (1-P_q)u(t) + m\mu p_m e^{-(1-\rho)m\mu t}.
\end{aligned}$$

This proves the theorem. ∎

Proof of Theorem 2. Since W is a random variable, $C(r, W)$, which is viewed as a function of W for a fixed r, is also a random variable. The expected charge to a service request with execution requirement r is (in the following, dt in parenthesis is the product of

the penalty factor and the time variable)

$$\begin{aligned}
&C(r)\\
&=\overline{C(r,W)}\\
&=\int_0^{\infty} f_W(t)C(r,t)dt\\
&=\int_0^{(a/d+c/s_0-1/s)r} f_W(t)C(r,t)dt\\
&=\int_0^{(a/d+c/s_0-1/s)r} ((1-P_q)u(t)\\
&\quad +m\mu p_m e^{-(1-\rho)m\mu t})C(r,t)dt\\
&=\int_0^{(a/d+c/s_0-1/s)r} (1-P_q)u(t)C(r,t)dt\\
&\quad +\int_0^{(a/d+c/s_0-1/s)r} m\mu p_m e^{-(1-\rho)m\mu t}C(r,t)dt\\
&=(1-P_q)ar+\int_0^{(c/s_0-1/s)r} m\mu p_m e^{-(1-\rho)m\mu t}C(r,t)dt\\
&\quad +\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} m\mu p_m e^{-(1-\rho)m\mu t}C(r,t)dt\\
&=(1-P_q)ar+\int_0^{(c/s_0-1/s)r} m\mu p_m e^{-(1-\rho)m\mu t}ardt\\
&\quad +\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} m\mu p_m e^{-(1-\rho)m\mu t}\\
&\qquad \left(\left(a+\frac{cd}{s_0}-\frac{d}{s}\right)r-dt\right)dt\\
&=(1-P_q)ar+m\mu p_m ar\int_0^{(c/s_0-1/s)r} e^{-(1-\rho)m\mu t}dt\\
&\quad +m\mu p_m\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} e^{-(1-\rho)m\mu t}\\
&\qquad \left(\left(a+\frac{cd}{s_0}-\frac{d}{s}\right)r-dt\right)dt\\
&=(1-P_q)ar+m\mu p_m ar\int_0^{(c/s_0-1/s)r} e^{-(1-\rho)m\mu t}dt\\
&+m\mu p_m\left(a+\frac{cd}{s_0}-\frac{d}{s}\right)r\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} e^{-(1-\rho)m\mu t}dt\\
&\quad -dm\mu p_m\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} te^{-(1-\rho)m\mu t}dt.
\end{aligned}$$

To continue the calculation, we notice that

$$\int e^{bt}dt=\frac{e^{bt}}{b},$$

and

$$\int te^{bt}=\frac{1}{b}\left(t-\frac{1}{b}\right)e^{bt}.$$

Hence, we have

$$\begin{aligned}
\int_0^{(c/s_0-1/s)r} e^{-(1-\rho)m\mu t}dt &= -\frac{e^{-(1-\rho)m\mu t}}{(1-\rho)m\mu}\Bigg|_0^{(c/s_0-1/s)r}\\
&= \frac{1-e^{-(1-\rho)m\mu(c/s_0-1/s)r}}{(1-\rho)m\mu},
\end{aligned}$$

and

$$\begin{aligned}
&\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} e^{-(1-\rho)m\mu t}dt\\
&=-\frac{e^{-(1-\rho)m\mu t}}{(1-\rho)m\mu}\Bigg|_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r}\\
&=\frac{e^{-(1-\rho)m\mu(c/s_0-1/s)r}-e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}}{(1-\rho)m\mu},
\end{aligned}$$

and

$$\begin{aligned}
&\int_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r} te^{-(1-\rho)m\mu t}dt\\
&=-\frac{1}{(1-\rho)m\mu}\left(t+\frac{1}{(1-\rho)m\mu}\right)\\
&\qquad e^{-(1-\rho)m\mu t}\Bigg|_{(c/s_0-1/s)r}^{(a/d+c/s_0-1/s)r}\\
&=\frac{1}{(1-\rho)m\mu}\Bigg(\left(\left(\frac{c}{s_0}-\frac{1}{s}\right)r+\frac{1}{(1-\rho)m\mu}\right)\\
&\qquad e^{-(1-\rho)m\mu(c/s_0-1/s)r}\\
&\quad -\left(\left(\frac{a}{d}+\frac{c}{s_0}-\frac{1}{s}\right)r+\frac{1}{(1-\rho)m\mu}\right)\\
&\qquad e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}\Bigg).
\end{aligned}$$

Based on the above results, we get

$$\begin{aligned}
&C(r)\\
&=(1-P_q)ar+m\mu p_m ar\frac{1-e^{-(1-\rho)m\mu(c/s_0-1/s)r}}{(1-\rho)m\mu}\\
&\quad +m\mu p_m\left(a+\frac{cd}{s_0}-\frac{d}{s}\right)r\\
&\qquad \frac{e^{-(1-\rho)m\mu(c/s_0-1/s)r}-e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}}{(1-\rho)m\mu}\\
&\quad -dm\mu p_m\frac{1}{(1-\rho)m\mu}\\
&\qquad \Bigg(\left(\left(\frac{c}{s_0}-\frac{1}{s}\right)r+\frac{1}{(1-\rho)m\mu}\right)e^{-(1-\rho)m\mu(c/s_0-1/s)r}\\
&\qquad\quad -\left(\left(\frac{a}{d}+\frac{c}{s_0}-\frac{1}{s}\right)r+\frac{1}{(1-\rho)m\mu}\right)\\
&\qquad\qquad e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}\Bigg)\\
&=(1-P_q)ar+\frac{ap_m}{1-\rho}\left(r-re^{-(1-\rho)m\mu(c/s_0-1/s)r}\right)\\
&\quad +\frac{p_m}{1-\rho}\left(a+\frac{cd}{s_0}-\frac{d}{s}\right)\\
&\qquad \left(re^{-(1-\rho)m\mu(c/s_0-1/s)r}-re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}\right)\\
&\quad -\frac{dp_m}{1-\rho}\Bigg(\left(\frac{c}{s_0}-\frac{1}{s}\right)re^{-(1-\rho)m\mu(c/s_0-1/s)r}\\
&\qquad +\frac{1}{(1-\rho)m\mu}e^{-(1-\rho)m\mu(c/s_0-1/s)r}
\end{aligned}$$

$$
\begin{aligned}
&- \left(\frac{a}{d} + \frac{c}{s_0} - \frac{1}{s}\right) re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \\
&- \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \Bigg) \\
&= (1-P_q)ar + aP_q\left(r - re^{-(1-\rho)m\mu(c/s_0-1/s)r}\right) \\
&\quad + P_q\left(a + \frac{cd}{s_0} - \frac{d}{s}\right) \\
&\quad \left(re^{-(1-\rho)m\mu(c/s_0-1/s)r} - re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}\right) \\
&\quad - dP_q\Bigg(\left(\frac{c}{s_0} - \frac{1}{s}\right) re^{-(1-\rho)m\mu(c/s_0-1/s)r} \\
&\qquad + \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(c/s_0-1/s)r} \\
&\qquad - \left(\frac{a}{d} + \frac{c}{s_0} - \frac{1}{s}\right) re^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \\
&\qquad - \frac{1}{(1-\rho)m\mu} e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r} \Bigg) \\
&= ar - \frac{dP_q}{(1-\rho)m\mu} \\
&\quad \left(e^{-(1-\rho)m\mu(c/s_0-1/s)r} - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)r}\right).
\end{aligned}
$$

Since r is a random variable, $C(r)$, which is viewed as a function of r, is also a random variable. Let the pdf of task execution requirement r to be

$$f_r(z) = \frac{1}{\bar{r}} e^{-z/\bar{r}}.$$

The expected charge to a service request is

$$
\begin{aligned}
&C \\
&= \overline{C(r)} \\
&= \int_0^\infty f_r(z)C(z)dz \\
&= \int_0^\infty \frac{1}{\bar{r}} e^{-z/\bar{r}} C(z)dz \\
&= \frac{1}{\bar{r}} \int_0^\infty e^{-z/\bar{r}} \Bigg(az - \frac{dP_q}{(1-\rho)m\mu} \\
&\qquad \left(e^{-(1-\rho)m\mu(c/s_0-1/s)z}\right. \\
&\qquad\qquad \left. - e^{-(1-\rho)m\mu(a/d+c/s_0-1/s)z}\right)\Bigg) dz \\
&= \frac{1}{\bar{r}} \Bigg(a\int_0^\infty ze^{-z/\bar{r}}dz \\
&\quad - \frac{dP_q}{(1-\rho)m\mu}\left(\int_0^\infty e^{-((1-\rho)m\mu(c/s_0-1/s)+1/\bar{r})z}dz\right. \\
&\qquad \left. - \int_0^\infty e^{-((1-\rho)m\mu(a/d+c/s_0-1/s)+1/\bar{r})z}dz\right)\Bigg).
\end{aligned}
$$

Since

$$\int_0^\infty ze^{-bz}dz = -\frac{1}{b}\left(z + \frac{1}{b}\right)e^{-bz}\Big|_0^\infty = \frac{1}{b^2},$$

and

$$\int_0^\infty e^{-bz}dz = -\frac{e^{-bz}}{b}\Big|_0^\infty = \frac{1}{b},$$

we get

$$
\begin{aligned}
&C \\
&= \frac{1}{\bar{r}}\Bigg(a\bar{r}^2 - \frac{dP_q}{(1-\rho)m\mu}\left(\frac{1}{(1-\rho)m\mu(c/s_0-1/s)+1/\bar{r}}\right. \\
&\qquad \left. - \frac{1}{(1-\rho)m\mu(a/d+c/s_0-1/s)+1/\bar{r}}\right)\Bigg) \\
&= a\bar{r} - \frac{dP_q}{(1-\rho)m\mu}\Bigg(\frac{1}{\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1} \\
&\qquad - \frac{1}{\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1}\Bigg) \\
&= a\bar{r} - \frac{dP_q}{(1-\rho)m\mu} \cdot \frac{\bar{r}(1-\rho)m\mu(a/d)}{(\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1)} \\
&\qquad \times \frac{1}{(\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1)} \\
&= a\bar{r} - \frac{a\bar{r}P_q}{(\bar{r}(1-\rho)m\mu(c/s_0-1/s)+1)} \\
&\qquad \times \frac{1}{(\bar{r}(1-\rho)m\mu(a/d+c/s_0-1/s)+1)} \\
&= a\bar{r}\Bigg(1 - \frac{P_q}{((ms-\lambda\bar{r})(c/s_0-1/s)+1)} \\
&\qquad \times \frac{1}{((ms-\lambda\bar{r})(a/d+c/s_0-1/s)+1)}\Bigg).
\end{aligned}
$$

The theorem is proven. ∎

3 Further Research Directions

Our investigation in this paper is only an initial attempt in this area. We would like to mention several further research directions.

- First, in a cloud computing environment, a multiserver system can be dynamically configured as a virtual cluster from a physical cluster, or a virtual multicore server from a physical multicore processor, or a virtual multiserver system from any elastic and dynamic resources. Our profit maximization problem can be extended to such virtual multiserver systems. To this end, a queueing model that accurately describes such a virtual multiserver system is required and needs to be developed. Such a model should be able to characterize a virtual multiserver system from a partially available physical system with deterministic or randomized availability.
- Second, our profit maximization problem can be extended to multiple heterogeneous multiserver systems of different sizes and speeds and application environments with total power consumption constraint. This is a multi-variable optimization problem, which is much more complicated than

the optimization performed for a single multiserver system in this paper. Such optimization has significant and practical applications in designing energy-efficient data centers.

- Third, when a multicore server processor is spatially divided into several multicore servers, our profit maximization problem can be defined for multiple multiserver systems. When the cores have a fixed speed, the optimization problem has a total server size constraint. When the cores have variable speeds, the optimization problem has a total server size constraint as well as a power consumption constraint.
- Fourth, when a physical machine is temporally partitioned into several virtual machines, i.e., when we are facing a dynamic cloud configuration with multi-tenant utilization, our profit maximization problem might be defined for multiple multiserver systems with total server speed constraint. Again, this part of the research relies on an accurate queueing model for virtual machines which is currently not available, although some effort has been made [9].

We believe that the effort made in this paper should inspire significant subsequent studies in profit maximization for cloud computing.

REFERENCES

[1] http://en.wikipedia.org/wiki/CMOS

[2] http://en.wikipedia.org/wiki/Service_level_agreement

[3] M. Armbrust, *et al*., "Above the clouds: a Berkeley view of cloud computing," Technical Report No. UCB/EECS-2009-28, February 2009.

[4] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," *Concurrency and Computation: Practice and Experience*, vol. 14, pp. 1507-1542, 2007.

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599-616, 2009.

[6] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal on Solid-State Circuits*, vol. 27, no. 4, pp. 473-484, 1992.

[7] B. N. Chun and D. E. Culler, "User-centric performance analysis of market-based cluster batch schedulers," *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

[8] D. Durkee, "Why cloud computing will never be free," *Communications of the ACM*, vol. 53, no. 5, pp. 62-69, 2010.

[9] R. Ghosh, K. S. Trivedi, V. K. Naik, and D. S. Kim, "End-to-end performability analysis for infrastructure-as-a-service cloud: an interacting stochastic models approach," *Proceedings of 16th IEEE Pacific Rim International Symposium on Dependable Computing*, pp. 125-132, 2010.

[10] K. Hwang, G. C. Fox, and J. J. Dongarra, *Distributed and Cloud Computing*, Morgan Kaufmann, 2012.

[11] Intel, *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor – White Paper*, March 2004.

[12] D. E. Irwin, L. E. Grit, and J. S. Chase, "Balancing risk and reward in a market-based task service," *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing*, pp. 160-169, 2004.

[13] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936-943, 2012.

[14] L. Kleinrock, *Queueing Systems, Volume 1: Theory*, John Wiley and Sons, New York, 1975.

[15] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profit-driven service request scheduling in clouds," *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 15-24, 2010.

[16] K. Li, "Optimal load distribution for multiple heterogeneous blade servers in a cloud computing environment," *Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium Workshops* (8th High-Performance Grid and Cloud Computing Workshop), pp. 943-952, Anchorage, Alaska, May 16-20, 2011.

[17] K. Li, "Optimal configuration of a multicore server processor for managing the power and performance tradeoff," *Journal of Supercomputing*, DOI: 10.1007/s11227-011-0686-1, published online 28 September 2011.

[18] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, 2009. http://csrc.nist.gov/groups/SNS/cloud-computing/

[19] F. I. Popovici and J. Wilkes, "Profitable services in an uncertain world," *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, 2005.

[20] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra: a computational economy-based job scheduling system for clusters," *Software – Practice and Experience*, vol. 34, pp. 573-590, 2004.

[21] C. S. Yeo and R. Buyya, "A taxonomy of market-based resource management systems for utility-driven cluster computing," *Software – Practice and Experience*, vol. 36, pp. 1381-1419, 2006.

[22] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," *Proceedings of the 41st Design Automation Conference*, pp. 868-873, 2004.