Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 756592, 8 pages
http://dx.doi.org/10.1155/2014/756592

Research Article

Dynamic Performance Optimization for Cloud Computing Using M/M/m Queueing System

Lizheng Guo,[1,2] Tao Yan,[1] Shuguang Zhao,[2] and Changyuan Jiang[2]

[1] *Department of Computer Science and Engineering, Henan University of Urban Construction, Pingdingshan, Henan 467036, China*
[2] *College of Information Sciences and Technology, Donghua University, Shanghai 201620, China*

Correspondence should be addressed to Lizheng Guo; kftjhpds@163.com

Received 28 August 2013; Revised 23 January 2014; Accepted 28 February 2014; Published 30 March 2014

Academic Editor: Guangchen Wang

Successful development of cloud computing has attracted more and more people and enterprises to use it. On one hand, using cloud computing reduces the cost; on the other hand, using cloud computing improves the efficiency. As the users are largely concerned about the Quality of Services (QoS), performance optimization of the cloud computing has become critical to its successful application. In order to optimize the performance of multiple requesters and services in cloud computing, by means of queueing theory, we analyze and conduct the equation of each parameter of the services in the data center. Then, through analyzing the performance parameters of the queueing system, we propose the synthesis optimization mode, function, and strategy. Lastly, we set up the simulation based on the synthesis optimization mode; we also compare and analyze the simulation results to the classical optimization methods (short service time first and first in, first out method), which show that the proposed model can optimize the average wait time, average queue length, and the number of customer.

1. Introduction

Cloud computing is a novel paradigm for the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the costs of management and maintenance of hardware and software resources [1]. This cloud concept emphasizes the transfers of management, maintenance, and investment from the customer to the provider. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2].

Generally speaking, cloud computing provides the capability through which typically real-time scalable resource like files, programs, data, hardware, computing, and the third party services can be accessible via the network to users. These users get the computing resources and services by means of customized service level agreement (SLA); they only pay the fee according to the using time, using manner, or the amount of data transferring. Any SLA management strategy consists of two well-differentiated phases: the negotiation of the contract and the monitoring of its fulfillment in real-time. Thus, SLA management encompasses the SLA contract definition: basic schema with the QoS parameters. In all the aspects, the QoS is the basis. of the cloud computing providing services to users QoS includes availability, throughput, reliability, and security, as well as many other parameters, but performance indicators are such as response time, task blocking probability, probability of immediate service, and mean number of tasks in the system [3], all of which may be determined by using the tool of queuing theory [4]. In order to agree with the QoS of the customers, thus, it is important to optimize the QoS. As cloud computing dynamically provides computing resource to meet the needs of QoS requesting from different customer, optimizing resource utilization will be a difficult task. On the other hand, a data center has a large number of physical computing nodes [5]; traditional queuing analysis rarely concerns systems of this size. Although several approaches have been proposed on critical research issues in cloud computing, including cloud security [6–8], privacy [9, 10], energy efficiency [11], and resource management

[12–14], optimizing researches with regard to performance are few.

In this paper, the data center is molded as a service center which can be used as an M/M/m queueing system with multiple tasks arrivals and task request buffer of infinite capacity. Through the M/M/m queueing theory, we deduce the equation of each parameter; then, we design an optimization function and a synthetical optimization method. Simulation results show that the proposed optimization method improves the performance of the data center compared with the classic method of short service time first and first-in, first-out.

The remainder of this paper is organized as follows. Section 2 discusses the related work on performance optimization and analysis. Section 3 gives the queueing model and optimization strategy. We present simulation setting and simulation results and then analyze and compare the results to other classical methods such as shorter time first and first in, first out in Section 4. Our works are summarized in Section 5, where we also outline the direction for future work.

2. Related Work

Although cloud computing has attracted research attention, only a small portion of the work has addressed the performance optimization question so far. In [15], Li put forward a differentiated service job scheduling system for a Cloud computing; then, by analyzing the differential QoS requirements of user jobs, he builds the corresponding non-preemptive priority M/G/1 queuing model for this system. They gave the corresponding algorithm to get the approximate optimistic value of service to each job with different priorities. In [16], the distribution of response time was obtained from using a cloud center model as the classic open network, assuming that both interarrival and service times are exponential. Using the distribution of the response time, the relationship among the maximum number of tasks, minimum service resources, and highest level of service was found. In [17], they used a linear predicting method and flat period reservation-reduced method to get useful information from the resource utilization log and made the M/M/1 queuing theory predicting method possess better response time and less energy-consuming. In [18], they employ the queuing model to investigate resource allocation problems in both single-class service case and multiple-class service case. Furthermore, they optimize the resource allocation to minimize the mean response time or minimize the resource cost in each case.

In addition, some researchers have undertaken the research of the performance analysis. In [19], the author proposed an M/G/m queuing system which indicates that interarrival time of requests is exponentially distributed; the service time is generally distributed and the number of facility nodes is m. In another paper [20], the authors have modeled the cloud center as an M/G/m/m + r queueing system with single task arrivals and a task request buffer of finite capacity. In order to evaluate the performance, they used a combination of a transform-based analytical model and an embedded Markov chain model, which obtained a complete probability distribution of response time and number of task in the system. Simulation results showed that their model and method provided accurate results for the mean number of tasks in the system, blocking probability, probability of immediate service as well as the response time distribution characteristics such as mean and standard deviation, skewness, and kurtosis.

In [21], the authors proposed an analytical queueing based model for performance management on cloud. In their research, the web applications were modeled as queues and virtual machines were modeled as service centers. They applied the queueing theory models to dynamically create and remove virtual machines in order to implement scaling up and down.

In [22], the authors analyzed the general problem of resource provisioning within cloud computing. In order to support decision making with respect to resource allocation for a cloud resource provider when different clients negotiated different service level agreements, they have modeled a cloud center using the M/M/C/C queueing system with different priority classes. The main performance criterion in their analysis was the rejection probability for different customer classes, which can be analytically determined.

From above analysis, we know that about the performance evaluation and analysis using the queueing theory has been researched in cloud computing, but with regard to the performance optimization researchs is rare. Moreover, as each of the parameters of the QoS has been studied in existing research, there is no work that addresses all of them simultaneously. In this paper, we use an M/M/m queueing system with multiple tasks arrivals and task request buffer of infinite capacity to optimize the performance.

3. Queueing Model for the Cloud Computing and Optimization Strategy

In cloud computing, there are a lot of users who access the service. We model cloud computing as in Figure 1. This model consists of cloud architecture which can be a service center. The service center is a single point of access for all kinds of customers all over the world. The service center is a collection of service resources which is provided by the provider to host all applications for users. Each user can apply to use the service according to the different kinds of requesting and pays some money to the provider of the service.

Cloud computing provider builds the service center to be used by customers, such as Amazon which provides several kinds of manners. In this paper, we use the on-demand instances. On-demand instances let you pay for compute capacity by the hour with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs [23].

The cloud computing service model displayed in Figure 1 can be mapped as a queueing model in Figure 2. Assuming that there are n requests and m services, each of them is independent. Since the consecutive arriving requests may be sent from two different users, the interarrival time is a

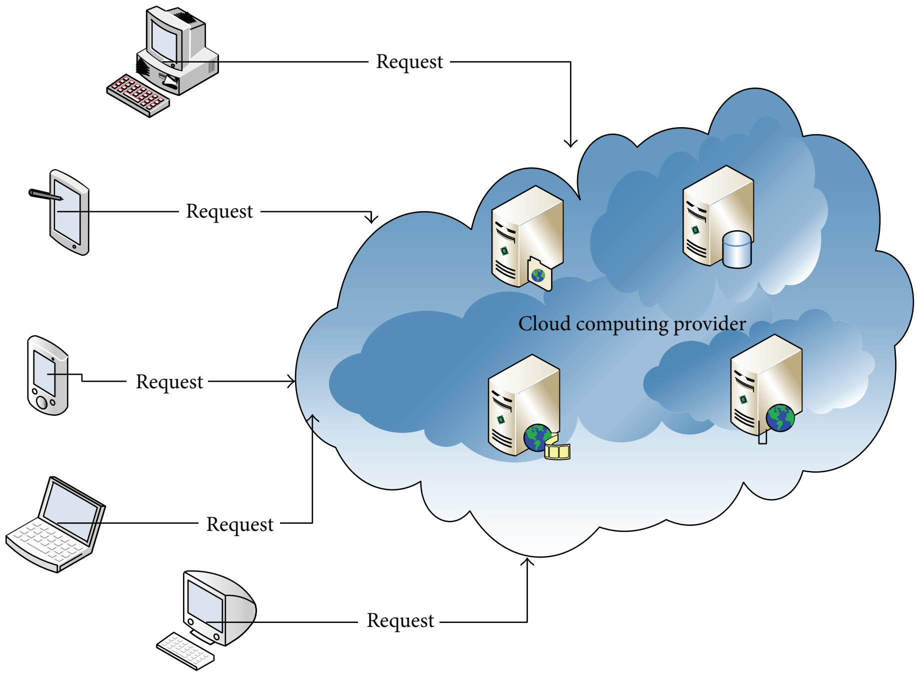

FIGURE 1: An illustration of request for cloud computing service model.

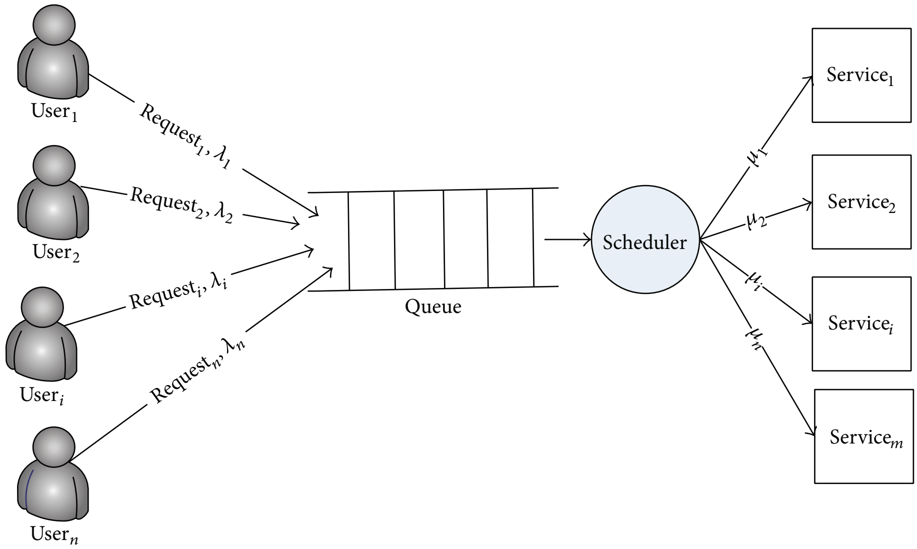

FIGURE 2: A queueing performance mode for computer services in cloud computing.

random variable, which can be modeled as an exponential random variable in cloud computing. Therefore, the arrivals of the requests follow a Poisson Process with arrival rate λ_i. Requests in the scheduler's queue are distributed to different computing servers and the scheduling rate depends on the scheduler. Suppose that there are m computing servers, denoted as Service$_1$, Service$_2$, Service$_i$, and Service$_m$ in the data center; the service rate is μ_i. So, the total arrival rate is $\lambda = \sum_{i=1}^{n} \lambda_i$ and the total service rate is $\mu = \sum_{j=1}^{m} u_i$. Theory has proved that the system is stable, when $\lambda/\mu < 1$. The rate of service requirement follows the Poisson Process; it is the same as the customer arriving rate. So, the M/M/m queuing model is fit for the cloud computing model.

3.1. Steady-State Equations. As the requests of the customers come from all over the world and the cloud computing can provide infinite services, the source of the customers and the number of the queuing model are not limited. The state set of the system is $E = \{0, 1, 2, \ldots\}$; so these balance equations can also be described by the state transition diagram of M/M/m shown in Figure 3.

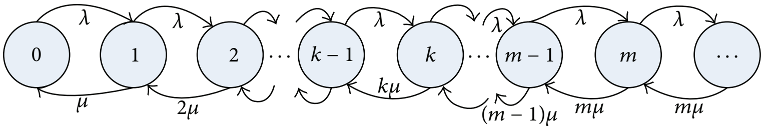

Figure 3: State-transition probability diagram.

When the state is $k(0 \le n \le m)$, the n services are busing and the other $m-n$ services are idle; $n > m$, where m service is busing and $n-m$ customers are waiting. Letting $\rho = \lambda/\mu$ and assuming the stability condition $\rho < 1$, the M/M/m queue gives rise to the following steady-state equations:

$$\begin{aligned} p_1 &= m\rho p_0, \\ p_2 &= \frac{m^2}{2!}\rho^2 p_0, \\ p_3 &= \frac{m^3}{3!}\rho^3 p_0, \\ p_m &= \frac{m^m}{m!}\rho^m p_0, \\ &\cdots \\ p_{m+r} &= \frac{m^m}{m!}\rho^{m+r} p_0. \end{aligned} \tag{1}$$

And, in general,

$$\begin{aligned} p_n &= \frac{m^n}{n!}\rho^n p_0, \quad 0 \le n < m, \\ p_n &= \frac{m^m}{m!}\rho^n p_0, \quad n \ge m. \end{aligned} \tag{2}$$

To obtain p_0, we sum up both sides of (2), and because the sum the $\sum_{n=0}^{\infty} p_n = 1$, we obtain an equation for p_0, in which its solution is

$$p_0 = \left(\sum_{n=0}^{m-1} \frac{\rho^n}{n!} + \frac{\rho^k}{k!} \times \frac{k}{k-\rho} \right)^{-1}. \tag{3}$$

3.2. Mean Queue Size, Delay, and Waiting Time. In order to evaluate and optimize the performance, we should deduce equation of the parameter. Firstly, we define the following notations:

L_s is a random variable representing the total number of customers in the system (waiting in the queue and being served);

L_q is a random variable representing the total number of customers waiting in the queue (this does not include those customers being served);

N_s is a random variable representing the total number of customers that are being served;

W_s is a random variable representing the total delay in the system (this includes the time a customer waits in the queue and in service);

W_q is a random variable representing the time a customer waits in the queue (this excludes the time a customer spends in service);

τ is a random variable representing the service time.

Using the above notations, we have

$$\begin{aligned} E[L_s] &= E[L_q] + E[N_s], \\ E[W_s] &= E[W_q] + E[\tau]. \end{aligned} \tag{4}$$

Clearly,

$$E[\tau] = \frac{1}{\mu}. \tag{5}$$

To obtain $E[N_s]$ for the M/M/m queue, we use Little's formula for the system. If we consider the system of servers (without considering the waiting room outside the servers), we notice that, since there are no losses, the arrival rate in this system is λ, and the mean waiting time of each customer in this system is $E[\tau] = 1/\mu$. Therefore, by Little's formula the mean number of busy servers is given by

$$E[N_s] = \frac{\lambda}{\mu} = \rho. \tag{6}$$

To obtain $E[L_q]$, we presume two mutually exclusive and exhaustive events: $\{q \ge m\}$, and $q < m$; we have

$$\begin{aligned} E[L_q] &= E[L_q \mid q \ge m] P(q \ge m) \\ &\quad + E[L_q \mid q < m] P(q < m). \end{aligned} \tag{7}$$

To get $E[L_q \mid q \ge m]$, we notice that the evolution of the M/M/m queue during the time when $q \ge m$ is equivalent to that of an M/M/1 queue with arrival rate λ, and service rate $m\mu$; so, the mean queue size of such M/M/1 queue is equal to $\rho/(1-\rho)$, where $\rho = \lambda/m\mu$. Thus,

$$E[L_q \mid q \ge m] = \frac{\rho/m}{1-\rho/m} = \frac{\rho}{m-\rho}. \tag{8}$$

Therefore, since $E[L_q \mid q < m] = 0$ and $P(q \ge m) = C_m(\rho)$, we obtain

$$E[L_q] = C_m(\rho)\frac{\rho}{k-\rho}. \tag{9}$$

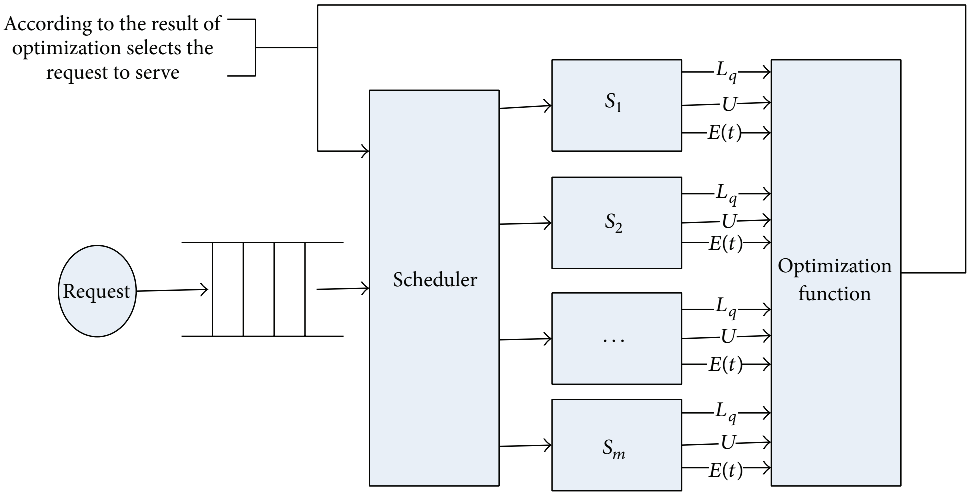

Figure 4: Task scheduling and performance optimization model based on queueing system.

3.3. Optimization Strategy and Optimization Algorithm. In cloud computing, the data center has many services. In Section 3.2, we have discussed the parameters of queueing in the cloud computing system. The parameters of the customer in the queueing are the L_s, L_q, and N_s; among the three parameters, the L_q plays the dominant role, because the customer being serviced is determined by the numbers on the server and the L_q is determined by the performance of the server. In the parameters of the service, we use the utilization rate, the mean time, and the number of customers waiting for servicing at each server as the optimization parameters. Task scheduling and performance optimization model based on a queueing system and service system are illustrated in Figure 4. The optimization strategy is as follows: all the requests arrive at the scheduler which selects the request to the server according to the result of the optimization function. The input parameter of the optimization function is the mean of the time, utilization, and the total number of customers waiting for each server in the cloud computing. Assuming that p_i is the probability of i customer to be selected and then to be executed; the i customer's servicing time is t_i. Therefore, the mean time $E(T)$ is as follows:

$$E[t] = \sum_{i=1}^{m} p_i * t_i. \tag{10}$$

In order to obtain the optimum result and keep the load balance, the optimization function is as follows:

$$\text{fcn} = \alpha * E[t] + \beta * L_q + \chi * U_i, \tag{11}$$

where α, β, and χ are the coefficients which can be gotten through training and $E[t]$, L_q, and U are the meantime, customers in the server, and the utilization rate of each server, respectively. The utilization rate is $U_i = \lambda_i/\mu_i$, where λ_i is arrival rate that the customer is selected by the optimization function and μ_i is the service rate of the S_i. The optimization function counts the function value according to the imputer parameters, and then sorts the value in ascending order. The scheduler selects the server to execute the service according to the results of the optimization function.

```
input: request_list, server_list
output: server_number
for server in server_list do
    $E(t)$ = getvalue$E$(servier_number)
    $L_q$ = getvalue$L_q$(server_number)
    $U = \lambda_i/\mu_i$
end
for server in server_list do
    function_value = $\alpha * E(t)_i + \beta * (L_q)_i + \chi * U_i$
end
ascending sort of the valued of function_value
getting the index of the first function_value
scheduler schedules the request to the select_server
```

Algorithm 1: Optimization algorithm.

According to the optimal strategy, the optimizing algorithm is described in Algorithm 1.

4. Simulation and Results Analysis

4.1. Simulation Setting. In order to validate the performance of our optimization strategy, we use the discrete event tool of the MATLAB. All of the experiments are tested on an AMD Phenom II X4 B95 3.0 GHz with 2 G RAM under a Microsoft Windows XP environment and all the experiments were implemented in MATLAB R2009b. We classify the requests into four kinds in accordance with the standard on-demand instances of Amazon. The request interarrival time follows an exponential distribution and the arrival rate mean is $\{\lambda_1 = 30, \lambda_2 = 20, \lambda_3 = 15, \lambda_4 = 12\}$. The service follows exponential distribution too and the service rate mean is $\{\mu_1 = 3, \mu_2 = 2.4, \mu_3 = 2, \mu_4 = 1.71\}$. The coefficient of the optimization function is $\{\alpha = 0.1, \beta = 0.4, \chi = 2\}$; all the test

Table 1: Average wait time.

Service number	FIFO	SSF	SO	
4	705.2	114.8	688.2	
20	475.5	43.42	304.6	
40	166.6	47.4	0.59	
60	2.64	2.64	48	0.025
			52	0.0010
80	0.06	0.06	60	0

Table 2: Average queue length.

Service number	FIFO	SSF	SO	
4	892.3	892.3	890.6	
20	607.3	607.3	398.1	
40	206.2	206.2	0.76	
60	3.39	3.39	48	0.032
			52	0.0013
80	0.077	0.077	60	0

Table 3: Amount of service customer.

Service number	FIFO	SSF	SO
4	101	101	102
20	667	667	1083
40	1517	1517	1848
60	1848	1848	1848
80	1848	1848	1848

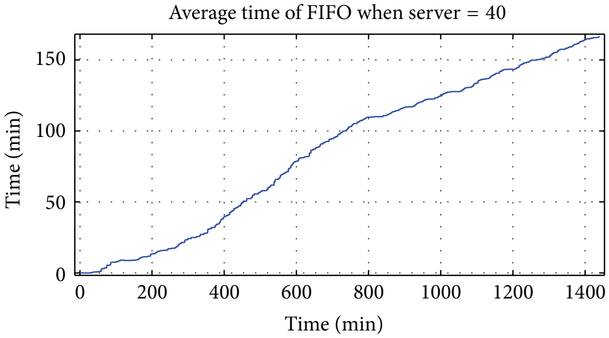

Figure 5: Average wait time in queue.

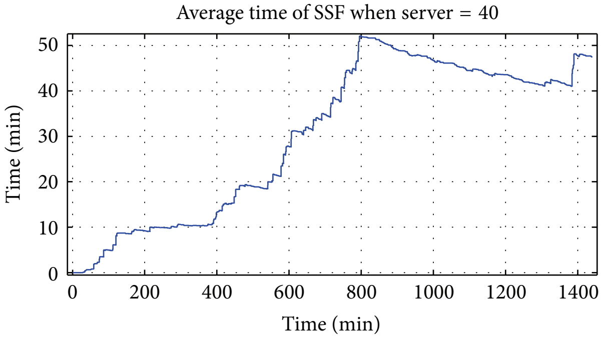

Figure 6: Average wait time in queue.

time is one day which is 1440 minutes; the number of servers is $\{4, 20, 40, 60, 80\}$.

4.2. Simulation Results and Analysis. We take our optimization method as a Priority Queue of synthesis optimization (SO). In order to compare and analyze the performance of optimization strategy, we use the classical Priority Queue of shorter service time first (SSF) and the first-in, first-out (FIFO) queuing policy as the test metric. We use the Priority Queue block to implement queuing policies based on optimization strategy and compare the performance of two prioritized queuing policies with the FIFO queuing policy.

Firstly, we test the average wait time when the servers are $\{4, 20, 40, 60, 80\}$, respectively. The results of three policies list are in Table 1. From Table 1, when the servers are 4 and 20, respectively, the SSF average wait time is the least; but, when the servers are 40, 60, and 80, the policy of the SO average time is the least. As the servers are little, many customers have to wait in queue; on the other hand, shorter services are firstly serviced and the service time of shorter services is less than that of other policies. So, the average wait time of the SSF policy is the best. However, when the service increases, that is, the service is 40, 60, and 80, the SO policy synthetically optimizes the utilization, average wait time, and average queue length and makes the best of each service capacity. Thus, the average wait time is shorter than that of other policies. Compared to SO policy, when the services increase, SSF and FIFO policy cannot consider the utilization of every service and queue length. Thus, some services may not fully work, and other services may overload, resulting in longer average time. On the other hand, the average queue length and the service customer number of three policies are shown in Tables 2 and 3. The above two tables represent the average queue length and the service customer number of SO is better than the other two policies. Although the average wait time of SSF is good when the service is 4 and 20, the average queue length and number of serviced customers of the SSF policy are not better than the SO policy. The reasons are as follows: the service is too little to provide enough service for the arrival customers and most of the customers have to wait in the queue. When the service number becomes more, the SO can optimize the average wait time, the average queue length, and the utilization. So, the SO gets more customers' accesses and the average queue length is shorter.

In order to clearly illustrate the average wait time of the queue, when the server is 40, the cure graph of the three policies is shown in Figures 5, 6, and 7. When customers requesting service are too many and the FIFO service policy cannot provide enough services, Figure 5 shows that the average wait time is almost linear increasing with the time going on. Although the wait time on average increases too, compared with Figure 5, Figure 6 has an obvious improvement. Figure 7 presents that, with the passage of time, the average wait time first goes up and then declines. The reason is as follows: due to the increase of users, firstly, the utilization rate of the server is not significantly improved, so the waiting

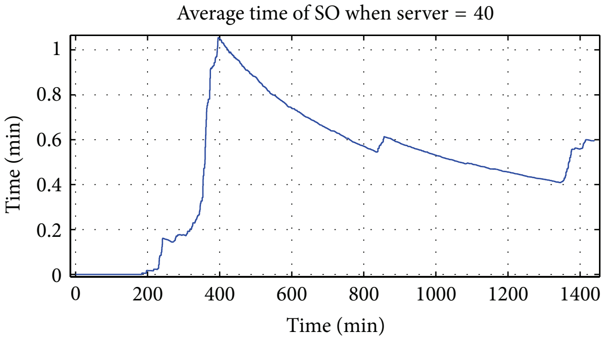

Figure 7: Average wait time in queue.

time will be increased; with the increase in the utilization rate of the server, the waiting time also reduces.

5. Conclusion and Future Work

We have studied the performance of the optimization and given the parameter of evaluating the service in cloud computing. Through the simulation, we can answer the following questions. (1) For given service numbers and customer's arrival rate, what level of QoS can be gained? (2) For given QoS requiring and customer's arrival rate, how many servers can meet the QoS? (3) For given service number and customer's arrival rate, how many customers can get the service? In this paper, in order to analyze the performance of services in cloud computing, we proposed a queueing model and developed a synthetical optimization method to optimize the performance of services. We have further conducted simulation to validate our optimization method. Simulation results showed that the proposed method can allow less wait time and queue length and more customers to gain the service. In order to evaluate the proposed systems in real cloud computing environment, we plan to implement it by extending real-world cloud platform, such as OpenStack. In addition, if the cloud computing is modeled as M/M/G, which can apply to diversity of service time. We will research this aspect in the future. At present, in cloud computing, data center power consumption is very huge; so another direction for future research is to optimize the performance and energy consumption. What is more, this work has social significance as it not only reduces the on-going operating costs but also decreases the carbon dioxide footprints.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been partially supported by the National Natural Science Foundation of China under Grant no. 61271114 and the Key Programs of Science and Technology Research of Henan Education Committee of China Grant no. 12A520006.

References

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[2] "The NIST Definition of Cloud Computing," http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[3] L. Wang, G. von Laszewski, A. Younge et al., "Cloud computing: a perspective study," *New Generation Computing*, vol. 28, no. 2, pp. 137–146, 2010.

[4] L. Kleinrock, *Queueing Systems: Theory*, vol. 1, Wiley-Interscience, New York, NY, USA, 1975.

[5] "Amazon Elastic Compute Cloud," User Guide, API Versioned. Amazon Web Service LLC or its affiliate, 2010, http://aws.amazon.com/documentation/ec2.

[6] W. Lu, A. L. Varna, and M. Wu, "Security analysis for privacy preserving search of multimedia," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 2093–2096, Hong Kong, September 2010.

[7] M. Menzel, R. Warschofsky, I. Thomas, C. Willems, and C. Meinel, "The service security lab: a model-driven platform to compose and explore service security in the cloud," in *Proceedings of the 6th IEEE World Congress on Services (Services-1 '10)*, pp. 115–122, Miami, Fla, USA, July 2010.

[8] D. Huang, X. Zhang, M. Kang, and J. Luo, "MobiCloud: building secure cloud framework for mobile computing and communication," in *Proceedings of the 5th IEEE International Symposium on Service-Oriented System Engineering (SOSE '10)*, pp. 27–34, Nanjing, China, June 2010.

[9] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 1–5, San Diego, Calif, USA, March 2010.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, San Diego, Calif, USA, March 2010.

[11] H. Yuan, C.-C. J. Kuo, and I. Ahmad, "Energy efficiency in data centers and cloud-based multimedia services: an overview and future directions," in *Proceedings of the International Conference on Green Computing*, pp. 375–382, Chicago, Ill, USA, August 2010.

[12] W. Lin and D. Qi, "Research on resource self-organizing model for cloud computing," in *Proceedings of the IEEE International Conference on Internet Technology and Applications (ITAP '10)*, pp. 1–5, Wuhan, China, August 2010.

[13] F. Teng and F. Magoulès, "Resource pricing and equilibrium allocation policy in cloud computing," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pp. 195–202, Bradford, UK, July 2010.

[14] H. Shi and Z. Zhan, "An optimal infrastructure design method of cloud computing services from the BDIM perspective," in *Proceedings of the 2nd Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA '09)*, pp. 393–396, Wuhan, China, November 2009.

[15] L. Q. Li, "An optimistic differentiated service job scheduling system for cloud computing service users and providers," in *Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering (MUE '09)*, pp. 295–299, Qingdao, China, June 2009.

[16] K. Xiong and H. Perros, "Service performance and analysis in cloud computing," in *Proceedings of the 5th IEEE World Congress*

on Services (Services-1 '09), pp. 693–700, Los Angeles, Calif, USA, September 2009.

[17] Y. Shi, X. Jiang, and K. Ye, "An energy-efficient scheme for cloud resource provisioning based on CloudSim," in *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER '11)*, pp. 595–599, Austin, Tex, USA, September 2011.

[18] X. M. Nan, Y. F. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proceedings of the 13th IEEE International Workshop on Multimedia Signal Processing (MMSP '11)*, pp. 1–6, Hangzhou, China, November 2011.

[19] H. Khazaei, J. Mišić, and V. B. Mišić, "Modelling of cloud computing centers using M/G/m queues," in *Proceedings of the 31st IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '11)*, pp. 87–92, Minneapolis, Minn, USA, June 2011.

[20] 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.

[21] V. Goswami, S. S. Patra, and G. B. Mund, "Performance analysis of cloud with queue-dependent virtual machines," in *Proceedings of the 1st IEEE International Conference on Recent Advances in Information Technology (RAIT '12)*, pp. 357–362, Dhanbad, India, 2012.

[22] W. Ellens, M. Zivkovic, J. Akkerboom, R. Litjens, and H. van den Berg, "Performance of cloud computing centers with multiple priority classes," in *Proceedings of the 5th IEEE International Conference on Cloud Computing (CLOUD '12)*, pp. 245–252, Honolulu, Hawaii, USA, 2012.

[23] Amazon EC2 Pricing, http://aws.amazon.com/ec2/pricing/.