Service Scheduling in Cloud Computing based on Queuing Game Model

Fuhong Lin[1], Xianwei Zhou[1], Daochao Huang[2], Wei Song[3] and Dongsheng Han[4]
[1] School of Computer and Communication Engineering, University of Science and Technology Beijing
Beijing, 100083, P. R. China
[e-mail: FHLin_ustb@yeah.net, XWZhouli@sina.com]
[2] National Computer Network Emergency Response Technical Team Coordination Center of China (CNCERT or CNCERT/CC)
Beijing 100029, P. R. China
[e-mail: huangdc@cert.org.cn]
[3]Research Institutes of Information Technology, Tsinghua University
Beijing 100084, P.R.China
[e-mail: wsong@mail.tsinghua.edu.cn]
[4]School of Electrical and Electronic Engineering, North China Electric Power University Baoding
Hebei 071003, P.R.China
[e-mail: dshhan@gmail.com]
*Corresponding author: Fuhong Lin

Received February 10, 2014; revised March 31, 2014; revised April 24, 2014; accepted April 26, 2014; published May 29, 2014

Abstract

Cloud Computing allows application providers seamlessly scaling their services and enables users scaling their usage according to their needs. In this paper, using queuing game model, we present service scheduling schemes which are used in software as a service (SaaS). The object is maximizing the Cloud Computing platform's (CCP's) payoff via controlling the service requests whether to join or balk, and controlling the value of CCP's admission fee. Firstly, we treat the CCP as one virtual machine (VM) and analyze the optimal queue length with a fixed admission fee distribution. If the position number of a new service request is bigger than the optimal queue length, it balks. Otherwise, it joins in. Under this scheme, the CCP's payoff can be maximized. Secondly, we extend this achievement to the multiple VMs situation. A big difference between single VM and multiple VMs is that the latter one needs to decide which VM the service requests turn to for service. We use a corresponding algorithm solve it. Simulation results demonstrate the good performance of our schemes.

Keywords: Cloud Computing, queue game, service scheduling, optimal queue length

http://dx.doi.org/10.3837/tiis.2014.05.003

1. Introduction

Cloud Computing can provide elastic resources which enables application providers seamlessly scaling their services. According to Buyya et al., [1] "a Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and the consumers". Also the goal of Cloud Computing is described by Chrysa et al., in [2] as "the goal of Cloud Computing is to create a fluid pool of virtual resources across computers, servers, and data centers that enable users to access stored data and applications on an as-needed basis". These statements explain how Cloud Computing works. According to the services they provided, the service model can be classified as the following three types, namely software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS).

SaaS belongs to a Service-Oriented Architecture, all the applications run on a Cloud Computing platform. Users' or clients' do not manage or control the underlying Cloud infrastructure including storage space, operating system, and network architecture. They just use a thin client device (e.g., smart phone, Tablet PC) which is a thin interface (e.g., web-based Brower) to get application services [3]. A hot research point in SaaS is how to scheduling the Cloud Computing platform resources among large number of users. In another words, if a user's application service request comes, how to schedule it (including whether or not serves it and which Virtual Machine (VM) serves it) in order to maximize the Cloud Computing's total payoff.

From the above statements, we know that the SaaS working mechanism can be treated as a queuing game problem. The main idea of queuing game is how to get equilibrium in the queue. The queuing game was firstly introduced by Naor in [4]. The subject of his idea is controlling a FCFS M/M/1 system. In his model, the customers react the queue manager who claims a constant admission fee by setting a pure strategy which is joining or balking the queue. Under this scheme all the customers, queue manager, and social aspect can maximize their payoffs.

This paper will focus on service scheduling schemes in Cloud Computing. While a new service request comes, it will be ranked in CCP's queue buffer, and then allocated it to a certain VM who becomes idle. In this situation, so the service request has to wait for a long time to get service since all the VMs are very busy. This is a waste of time. Hence, we design that the newly came service request decides whether to join or balk the buffer queue according to some factors. We simplify a Cloud Computing architecture into three parts, namely Clients or users' (C), Job Scheduler (JS) and VMs. The working mechanism is that the JS schedules the service request generated by Cs among VMs. Then single VM pricing scheme is proposed. We treat the Cloud Computing Platform (CCP) as one serving VM, and analyze the optimal queue length based on game theory with a fixed admission fee distribution. Using the obtained optimal queue length, we get the CCP's maximized payoff. After solving the single VM scheme, we pay attention to the multiple VMs scheme. A primary difference is that the JS needs to schedule which VM the service request to turn to for service. We propose an algorithm to solve this problem, and the main idea is that assuming the newly arrived service request will join in a certain VM's queue, and then it can be processed using single VM pricing scheme. All the possible VMs' payoffs are collected. Lastly, the service request will turn to the certain VM who has the biggest payoff. In another words, the JS needs to schedule the service

request to this VM. If no VM can provide a positive payoff, the service request will balk. To show the Cloud Computing platform performance under proposed schemes, we give three numerical simulations. The first one is analyzing the relationship between the optimal queue length and the biggest admission fee, the second one is analyzing the Cloud Computing platform's payoff under different distribution of admission fee, and the third one is analyzing the Cloud Computing platform's payoff under different service time and arrival rate.

The rest of this paper is organized as follows. Section 2 introduces some related work. Problem formulation and corresponding solution are given in Section 3. The numerical simulation is done in Section 4. And in Section 5, a conclusion is drawn.

2. Related Work

Since the Cloud Computing being proposed, fruitful outcomes are achieved in the research area of resource scheduling and allocation. There are mainly two types of work which are related to our work. The first one is Service Level Agreement (SLA) which is used a formal contract between users and Cloud servers to manage resources. The second on is Resource Provisioning which involves the suitable provisioning of Cloud resources (Such as CPU, memory, and bandwidth) to make the application to meet its QoS requirements with the time-varying workloads. We introduce some related work on these topics in [5-7] and [8-10] respectively. Also there are some researches combining these two types, and we showed them in [11, 12].

In [5], authors researched on the dynamic Cloud resource management. They claimed that manual management of Cloud systems is a challenging issue, for it grows bigger and bigger. SLA can manage large scale infrastructure management and support multiple dynamic requirement. So they introduced a SLA-aware PaaS Cloud platform which is used to manage Cloud resource lifecycle. This platform has the following features. It enables Cloud providers to deal with higher-level metrics, nearer to end-user perception, and with flexible composition of multiple actors' requirements. And it can dynamically adapt to guarantee the QoS requirements. The claimed that their scheme could achieve minimum cost and maximum efficiency via simulation considering several realistic workload profiles.

In [6], authors pointed out that owing to system malfunctions, hard- and software failures, changing workload conditions, SLAs in Cloud Computing can be violated. In oder to avoid this violation. So the SLA attainment strategies should be flexible and adaptive. Foucsing on distributed, heterogeneous, diverse and virtualized world of services, they presented a self-manageable SLA architecture to ease interoperable service executions. They ran a general biochemical application in CloudSim to validate the proposed scheme. The outcome fulfilled the expected utilization gains.

In [7], authors claimed that existing hypervisor schedulers might violate customers' SLAs in Cloud Computing, for the schedulers can not control the VM execution sequence and have to reduce the consolidation ratio to meet SLAs. So they proposed a system which can achieve SLAs while not penalize the consolidation ratio. They used two elements, namely Sage which is used as VM admission controller, and Shift which is used as a hypervisor scheduler. Based on incoming requests' patterns, Sage assesses its SLA easily. According to the amount of resource allocation and the sequence of VM execution, Shift maintains the admitted SLAs. Simulation results showed that their scheme could improve the consolidation ratio up to 66% without VM performance degradation

In [8], authors analyzed Cloud platforms' resource provisioning. They concluded that current Cloud environments have some challenges, such as live migration, quality of service and fault tolerance. Embedding of applications' and users' behaviour in the Cloud management processes could be useful for accurate resource provision estimation. So they proposed a two-level generic black-box scheme. This scheme can estimates resource attributes at a low level using the information at a high level which is related to application terms and it predicts the user behaviour. Information of high-level is abstrated through a time series analysis, and then being translated to low-level via Artificial Neural Networks. Lastly they implemented their scheme in different application scenarios to verify the effectiveness.

In [9], authors researched on the coordinated resource provisioning and maintenance scheduling scheme Cloud environments. They said that a third to a half downtime events are caused by lack of proper maintenance. A solving choice is to design an appropriate maintenance schedule. So they used joint VM resource provisioning and maintenance scheduling strategy to make the revenue be maximized. Firstly, they proposed a resource provisioning heuristic. Then they proposed an algorithm to solve the built problem and got its upper bound. Simulations resluts demonstrated that proposed heuristic algorithms could work effectively to maximize the Cloud revenue.

In [10], authors analyzed adaptive Cloud resource provisioning using empirical prediction models. They stated that a new virtual instance can not be instantaneously initialized. The Cloud plateform needs several minutes delay to allocate the hardware resource. So using Neural Network and Linear Regression, they proposed prediction-based resource measurement and provisioning strategies in order to satisfy upcoming resource demands. Simulations resluts showed that proposed technique assures a more adaptive resource management.

In [11], authors pointed out that SLAs could help resources provisioning. But during application provisioning, the SLA could be violated easily. And there was no proper solutions. So they proposed a framework called low-level metrics to high-level SLA to manage the monitoring of low-level resource metrics. Futher maps them to high-level SLAs. Using monitored information and SLA violation prevention techniques, this framework provided the application deployment mechanism. They tested their scheme in a real Cloud environment and the performance was good.

In [12], considering SLA, authors researched on the resource provisioning problem in Multi-Cloud Systems. They used a combinatorial auction-based approach. The aim was effectively provising and allocating dynamic resources in the condition of the client requests with deadline. Simulations resluts demonstrated that their auction-based approach could be effective.

In all, the above researches work on resource scheduling and allocation in Cloud environment. Comparing with them, our work concentrates on the user part. Controlling the service requests whether to join or balk and the value of CCP's admission fee, we maximize the CCP's payoff.

3. Problem Formulation and Solution

In this section, we firstly introduce the working mechanism of Clouding Computing and formulating the research problem. Then we propose the single VM pricing scheme for maximizing the CCP's payoff. Lastly, we research on the services scheduling and pricing scheme in the condition of multiple VMs.

3.1 Architecture

As shown in **Fig.1**, there are two entities in a Cloud Computing System, namely C and CCP. The Cs generate service requests and CCPs solve these request. The CCP can be further divided into JS and VMs. The JS takes in charge the service requests scheduling while VMs solve the assigned service requests. Upon the above statements, we give the following definition.

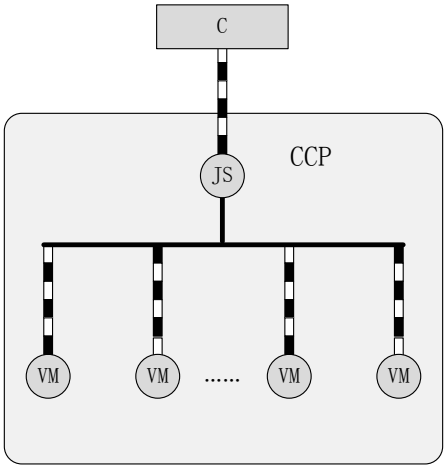


Fig. 1. Structural overview of Cloud Computing System

Definition 1: In a CCP, the service requests generated by clients are allocated by the JS among the VMs. Each VM can announce an admission fee in order to maximize its payoff. Each service request can decide whether to join or balk. And if it decides to join, it needs to choose which VM for service.

From this definition, we can formulate the research problem which is optimally assign the service requests among VMs in the condition of taking into account of the admission fee so as to maximize the CCP's total payoff. Parameters used in our model are as follows. The number of VM is $N(N \in N^+)$. The Clients' service arrival rate follows Poisson distribution with parameter λ. The service time of Cloud Computing VMs follows exponentially distribution with parameter μ $(\lambda < \mu)$. So the load level ρ equals to λ / μ. The $i^{'}$ th $i \in [1, M], M \in (N^+)$ client's benefit from completed service is R, and its cost from staying in the system per unit of time is C. The probability of a service joining the $j^{'}$ th VM is P_j.

3.2 Single VM Pricing Scheme

In this subsection, we consider the following Scenario. There is only one VM in the CCP. So when a service request comes, the JS can only assign it to this VM. Of course this is a simplified scheme compared with the traditional one. In this Scenario, we only need to research on how the VM announces the admission fee and whether or not the service request decide to accept the service under the announced admission fee.

From the above statements, we can get that this is an M/M/1 queuing game issue. Naor in [4] solved this problem under constant admission fee θ. He got that the optimal queue length is n_{so}, and the corresponding optimal admission fee is θ.

$$n_{so} = \lfloor \nu_{so} \rfloor = \left\lfloor \arg\left(\frac{R\mu}{C} - \nu_{so} - \frac{(1-\rho^{\nu_{so}-1})(1-\rho^{\nu_{so}+1})}{\rho^{\nu_{so}-1}(1-\rho)^2}\right) \right\rfloor \tag{1}$$

$$\theta = R - n_{so}\frac{C}{\mu} \tag{2}$$

Under these parameters, Refael in [13] got the server's optimal payoffs is P_{so}.

$$P_{so} = \lambda \frac{1-\rho^{\nu_{so}}}{1-\rho^{\nu_{so}+1}}(R - n_{so}\frac{C}{\mu}) \tag{3}$$

In our opinion, it is not suitable for the admission fee being constant. JS schedules the queued service request. If a service request comes and finds its rank is near the VM, then it should be levied a lower admission fee, for it can obtain service more quickly. Contrary, if the service request finds its rank is far from the VM, then it should be levied a higher admission fee, for it needs to wait more times to obtain service. So we propose that the admission fee for a newly arrived service request at the position k should be $\theta(k)$ which is a non-decreasing function. Under this assumption, we will research on the optimal length of the queue and VMs' maximized payoff.

Theorem 1: In a Single VM Cloud Computing platform, the threshold of a certain service request whether to join or balk is:

$$l_o, (l_o \in N^+ \,\&\, l_o \in [\frac{(R-\theta(l_o))\mu}{C} - 1, \frac{(R-\theta(l_o))\mu}{C}]) \tag{4}$$

Proof 1: The gain of i'th service request is equal to

$$G_i = R - \frac{(i+1)C}{\mu} - \theta(i) \tag{5}$$

Where R is the earn if it is being served by CCP, $\frac{(i+1)C}{\mu}$ is the cost of queue period, and $\theta(i)$ is the cost charged by CCP.

If the value of the gain is positive, the service request tends to accept the VM service, and join in the VM queue. If this value is zero, whether join or not makes no difference for the service request. Lastly, if this value is negative, the service request tends to reject the VM service, for it can not make any positive payoff. So the optimal length of the service request queue l_o should satisfy the following inequalities.

$$\begin{cases} G_{l_o} = R - \frac{l_o C}{\mu} - \theta(l_o) \geq 0 \\ G_{l_o} = R - \frac{(l_o+1)C}{\mu} - \theta(l_o) < 0 \end{cases} \tag{6}$$

Solving these inequalities, we can get

$$\begin{cases} l_o < \dfrac{(R-\theta(l_o))\mu}{C} - 1 \\ l_o \le \dfrac{(R-\theta(l_o))\mu}{C} \end{cases} \tag{7}$$

The value queue length should be an integer, and there is one and only one integer in the inequalities in (7), so the length of the service request queue l_o can be written as:

$$l_o, (\quad l_o \in N^+ \,\&\, l_o \in (\frac{(R-\theta(l_o))\mu}{C} - 1, \frac{(R-\theta(l_o))\mu}{C}] \quad) \tag{8}$$

Theorem 2: In a Single VM Cloud Computing platform, $p_i, i \in [1, l_o]$ represents the probability that there are i service requests in the VM serving queue. The VM's payoff can be maximized according to the proposed admission fee $\theta(k)$. And the maximized fee is:

$$G_{VM} = \lambda(1-p_{l_o})E(\theta(i)) = \lambda\frac{1-\rho^{l_o}}{1-\rho^{l_o+1}}(p_1\theta(1) + p_2\sum_{i=1}^{2}\theta(i) + p_3\sum_{i=1}^{3}\theta(i) + ... + p_{l_o}\sum_{i=1}^{l_o}\theta(i)) \tag{9}$$

Proof 2: Following the Theorem 1, the M/M/1 queue model turns to an M/M/1/l_o queue mode, the VM's gain can be written as

$$G_{VM} = \lambda(1-p_{l_o})E(\theta(i)) = \lambda\frac{1-\rho^{l_o}}{1-\rho^{l_o+1}}(p_1\theta(1) + p_2\sum_{i=1}^{2}\theta(i) + p_3\sum_{i=1}^{3}\theta(i) + ... + p_{l_o}\sum_{i=1}^{l_o}\theta(i)) \tag{10}$$

Where $p_i, i \in [1, l_o]$ is the probability that there are i service requests in the VM serving queue. According to the queue theory, it can be written as:

$$p_i = \frac{1-\rho}{1-\rho^{l_o+1}}\rho^i \quad i = 1, 2, ..., l_o \tag{11}$$

The different distribution of θ will generate different G_{VM}. We will analyze this characteristic in the numerical simulation method in the simulation section.

3.3 Multiple VMs Scheduling and Pricing Scheme

In this subsection, we further consider the following Scenario. There are n VMs in the CCP. When a service request comes, the JS schedules this request among VMs. Compared with the single VM scheme, the difference is that the service request can not only decide whether to join or balk, but also it can decide which VM it wants to turn to in order to get service. We assume that the $j^{'}$ th VM announced admission fee $\theta(j,k)$ for the position k. $\theta(j,k)$ is non-decreasing with the increasing of queue length.

From the subsection 3.2, we can get that for a certain VM queue j, the longest length of the service request queue should be

$$l_{j,o},(l_{j,o} \in N^{+} \,\&\, l_{j,o} \in (\frac{(R-\theta_j(l_{j,o}))\mu}{C}-1,\frac{(R-\theta_j(l_{j,o}))\mu}{C}]) \tag{12}$$

Also, we can get that when a certain service request comes at the position k, the VM $j^{'}$ payoff the can be written as:

$$G_{VM,j} = \theta_j(1)+\theta_j(2)+...+\theta_j(k) \tag{13}$$

Then the service request's joining procedure can be expressed as follows which is also shown in **Algorithm 1.** When a service request comes, it compares the value of his position k with every VM's optimal queue length. If this value is smaller, it stores the corresponding VM in the VM set S_{VM}. After comparing, the service request checks its VM set S_{VM}. If the S_{VM} is empty, the service request balks. Otherwise, it ranks all the VMs' payoff $G_{VM,j}$ in S_{VM}. And it chooses the one with maximum $G_{VM,j}$ for service.

This algorithm can make a local optimization for the Cloud Computing platform when a service request comes. Because of the additive characteristic, we conclude that Cloud Computing platform can achieve the final optimal. In other words, this scheme can maximize the Cloud Computing platform's total payoff.

Algorithm 1 the service request joining policy

For j =1:n
Calculate the $j^{'}$ th VM's optimal queue length $l_{j,o}$
  if $l_{j,o} < k$
    Store the $j^{'}$ th VM in S_{VM}
  else
    Store the $j^{'}$ th VM in S_{VM}^{-}
  end if
end for
if length (S_{VM}) = 0
  The service request balks
end if
for j =1:length (S_{VM})
  Calculate the $j^{'}$ th VM's payoff $G_{VM,j}$, and store it in G_{VM}
end for
$k = arg(\text{maximize}(G_{VM}))$
The service request chooses the $k^{'}$ th VM, and join its queue

4. Numerical Simulation and Analysis

In this section, some numerical simulations are carried out to show good performance of our proposed scheme in CCP. As described in the formal Section, the VMs set the admission fee, and the service request decides whether to join or balk to maxmize the CCP's payoff. So our scheme is different from the related ones. We will give a performace evalution in our designed Scenarios.

Three scenarios for performance demonstrating in single VM scheme are designed and analayzed by Matlab. The first one is analyzing the relationship between the optimal queue length and the biggest admission fee, the second one is analyzing the Cloud Computing platform's payoff under different distribution of admission fee, and the third one is analyzing the Cloud Computing platform's payoff under different service time and arrival rate. In the Multiple VMs case, using the outcome of single VM scheme and algorithm 1, the optimal payoff can be achieved, so we no not simulate it. For simpleness, we set a small scale Cloud network. The common parameters used are shown in **Table 1**. We can see that there are 5 servers in CCP with the arrival rate being 4 and the serving rate being 6. The benefit and cost of a service request are 10 and 2 respectively.

Table 1. Parameters used

N	λ	μ	R	C
5	4	6	10	2

In the first Scenario, we analyze show the relationship between the optimal queue length l_o and the maximum admission fee $\theta_{\max}(l_o)$ in single VM scheme. The value of $\theta_{\max}(l_o)$ is set being increasing from 0 to 5 with the step of 0.5. From **Fig. 2** we can see that the optimal queue length l_o decreases with the increasing of maximum admission fee $\theta_{\max}(l_o)$. The explanation is that if the VM charges an admission fee, each of the service request's payoff will be reduced. So there will be fewer service requests, who has a positive profit, ranking in the CCP' queue buffer.

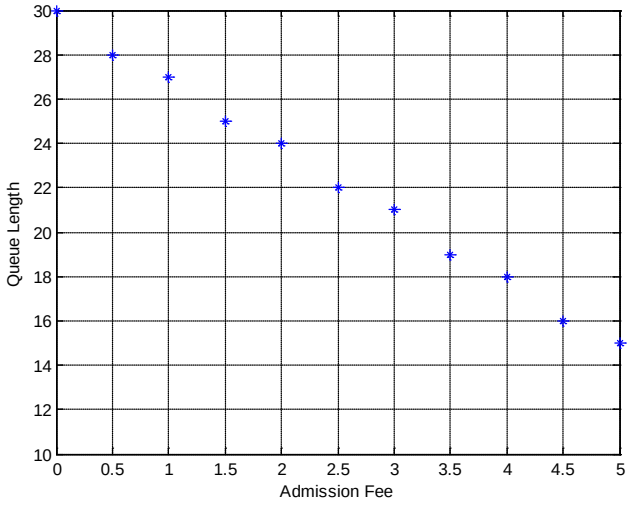


Fig. 2. The relationship between l_o and $\theta_{\max}(l_o)$

In the second Scenario, we analyze the Cloud Computing platform's payoff under different distribution of admission fee. As shown in **Fig. 3 (b)**, the admission fee charged by a certain VM of its different position follows y = x, y = x^2, y = x^3, y = x^4 respectively. Where x represents the position in the VM's buffer queue, and y represents the corresponding admission fee value. Parameters used in this Scenario are shown in Table 1 except the maximum admission fee $\theta_{\max}(l_o)$ being 5. From **Fig. 3 (a)** we can get that the Cloud Computing platform's payoff or gain is strongly affected by the corresponding admission fee's distribution, and the bigger this distribution's square deviation is, the smaller gain the Cloud Computing platform achieves. So we should choose the distribution which has a big square deviation as our admission fee distribution.

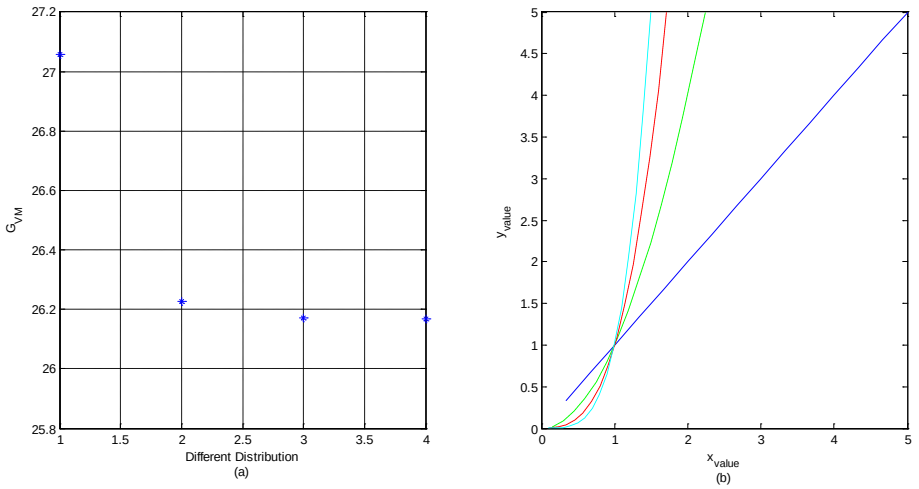


Fig. 3. The Cloud Computing platform's payoff under different distribution of admission fee

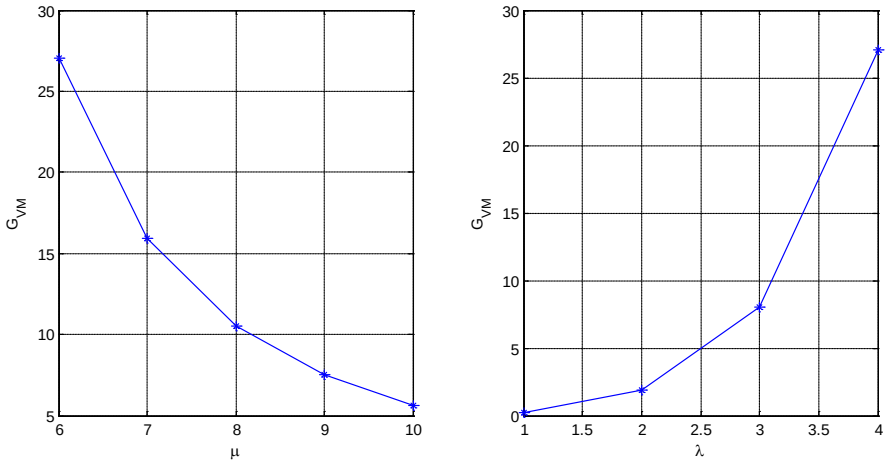


Fig. 4. The Cloud Computing platform's payoff under different service time and arrival rate

In the third Scenario, we analyze the CCP's payoff under different service time μ and arrival rate λ. From **Fig. 4** we can get that the increase of service time μ will decrease the

CCP's payoff. The reason is that increasing the service time will decrease the number of service requests being completely served in a unit of time. Also from **Fig. 4** we can get that the increasing of arrival rate λ will increase the CCP's payoff. The reason is that increasing the arrival rate will reduce the probability of VM being idle.

5. Conclusion

In this paper, service scheduling in Cloud Computing was analyzed using queuing game. We aimed at maximizing the Cloud Computing platform's total payoff via controlling the service scheduling strategy. When a user's service request came, it would decide whether to join or balk the queue according to the game theory. We analyzed two Scenarios. The first was that the Cloud Computing platform served as a whole using one VM. This VM could announce a different admission fee to different position in the queue. Under this condition, we could get the optimal queue length. If the position number of a new arrival service request is smaller than the optimal queue length, it joins in. Otherwise, it balks. Then we could get the maximized payoff of CCP. Further, we extended this achievement to the multiple VMs situation. Multiple VMs brought in a new problem which was that the service request needed to decide which VM the service requests turning to for service. A solving method was proposed using an algorithm, and the main idea of choosing the certain VM who could maximize the CCP's payoff. Three numerical simulations were put forward. The first one analyzed the relationship between the optimal queue length and the biggest admission fee, the second one analyzed the Cloud Computing platform's payoff under different distribution of admission fee, and the third one analyzed the Cloud Computing platform's payoff under different service time and arrival rate.

For the future, an aspect that is worth pursuing is the dynamic queue scheme. In the CCP part, one can build dynamic models of resource provision and pricing in order to maximize the CCP's payoff. In the user part, one can also consider some more applied models. For example, some impatient users who are already in the queue maybe balk it when the service efficiency is low. This case also affects the CCP's profit.

Acknowledgments

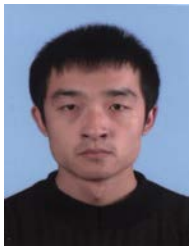
We gratefully acknowledge anonymous reviewers who read drafts and made many helpful suggestions. This work is supported by National Science Foundation Project of P. R. China (61202079), China Postdoctoral Science Foundation (2013M530526), the Fundamental Research Funds for the Central Universities (FRF-TP-09-015A) and Future Network Prospective Research Project of Jiangsu Science and Technology Department (No. BY2013095-2-14).

References

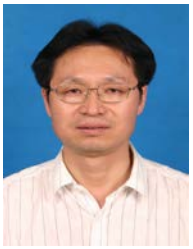
[1] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities," in *Proc. IEEE Int'l Conf. High Performance Computing and Communications (HPCC '08)*, pp. 5-13, Sept. 2008.
Article (CrossRef Link).

[2] Chrysa A. Papagianni, Aris Leivadeas, Symeon Papavassiliou, Vasilis Maglaris, Cristina Cervello-Pastor, Álvaro Monje, "On the Optimal Allocation of Virtual Resources in Cloud Computing Networks," *IEEE Trans. Computers (TC)*, 62(6):1060-1071, 2013.

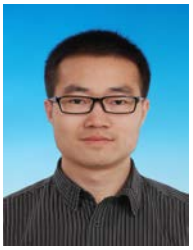
Article (CrossRef Link).
[3] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing," *National Institute of Standards and Technology, Special Publication*, 800-145, 2011. Article (CrossRef Link).
[4] P. Naor, "The regulation of queue size by levying tolls," *Econometrica*, vol. 37, pp.15-24, 1969. Article (CrossRef Link).
[5] Andrés García-García, Ignacio Blanquer Espert, Vicente Hernández García, "SLA-driven dynamic Cloud resource management," *Future Generation Comp. Syst. (FGCS)*, 31:1-11, 2014. Article (CrossRef Link).
[6] Attila Kertész, Gabor Kecskemeti, Ivona Brandic, "An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments," *Future Generation Comp. Syst. (FGCS)*, 32:54-68, 2014. Article (CrossRef Link).
[7] Orathai Sukwong, Akkarit Sangpetch, Hyong S. Kim, "SageShift: Managing SLAs for highly consolidated Cloud," *INFOCOM 2012*, 208-216, 2012. Article (CrossRef Link).
[8] George Kousiouris, Andreas Menychtas, Dimosthenis Kyriazis, Spyridon V. Gogouvitis, Theodora A. Varvarigou, "Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in Cloud platforms," *Future Generation Comp. Syst. (FGCS)*, 32:27-40, 2014. Article (CrossRef Link).
[9] Zeyu Zheng, Minming Li, Xun Xiao, Jianping Wang, "Coordinated resource provisioning and maintenance scheduling in Cloud data centers," *INFOCOM 2013*, 345-349, 2013. Article (CrossRef Link).
[10] Sadeka Islam, Jacky Keung, Kevin Lee, Anna Liu, "Empirical prediction models for adaptive resource provisioning in the Cloud," *Future Generation Comp. Syst. (FGCS)*, 28(1), 155-162, 2012. Article (CrossRef Link).
[11] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Schahram Dustdar, "Cloud resource provisioning and SLA enforcement via LoM2HiS framework," *Concurrency and Computation: Practice and Experience (CONCURRENCY)*, 25(10), 1462-1481, 2013. Article (CrossRef Link).
[12] Chi-Chung Chang, Kuan-Chou Lai, Chao-Tung Yang, "Auction-Based Resource Provisioning with SLA Consideration on Multi-Cloud Systems," *COMPSAC Workshops* 2013, 445-450, 2013. Article (CrossRef Link).
[13] R. Hassin and M. Haviv, "To Queue or Not to Queue," *Kluwer Academic Publishers*, 2003. Article (CrossRef Link).



Fuhong Lin received his M.S. degree and Ph.D degree from Beijing Jiaotong University, Beijing, China, in 2006 and 2010, respectively, both in Electronics Engineering. Now he is a lecturer in department of Computer and Communication Engineering, University of Science and Technology Beijing, P. R. China. His research interests include wisdom network, social network and multimedia network. Email: FHLin_ustb@yeah.net



Xianwei Zhou received his B.S. degree in Department of Mathematics from Southwest Normal University in 1986 and his M.S. degree from Zhengzhou University in 1992, and in 1999, he obtained Ph.D. degree in Department of Transportation Engineering from Southwest Jiaotong University, China. He was engaged in postdoctor study at Beijing Jiaotong University, China, from 1999 to 2000. Now, as a professor in Department of Communication Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, his research interests include the security of communication networks, next generation networks, scheduling and game theory.



Daochao Huang received his B.S. degree and Ph.D degree from Beijing Jiaotong University, Beijing, China, in 2007 and 2012, respectively, both in Electronics Engineering. His research interests are in the areas of communication networks including Cloud Computing, data center and Next Generation Internet technologies.

Wei Song received his M.S. degree and Ph.D degree from Beijing Jiaotong University, Beijing, China, in 2006 and 2010, respectively, both in Electronics Engineering. Now he is a lecturer in school of Information Engineering, Minzu University of China. He is also an assistant professor in Research Institutes of Information Technology, Tsinghua University. His research interests include multimedia network and image processing.

Dongsheng Han was born in Heilongjiang, China in 1980. He received the B.Eng. degree in Telecommunications Engineering from North China Electric Power University of China in 2003 and the Ph.D. degree in Communication and Information System from Beijing Jiaotong University of China in 2012. He is currently a lecturer in the School of Electrical and Electronic Engineering, North China Electric Power University Since Apr. 2012. His main research interests are digital communication system and wireless communication.