This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

Customer-Satisfaction-Aware Optimal Multiserver Configuration for Profit Maximization in Cloud Computing

Jing Mei, Kenli Li, *Member, IEEE* and Keqin Li, *Fellow, IEEE*

Abstract—Along with the development of cloud computing, an increasing number of enterprises start to adopt cloud service, which promotes the emergence of many cloud service providers. For cloud service providers, how to configure their cloud service platforms to obtain the maximum profit becomes increasingly the focus that they pay attention to. In this paper, we take customer satisfaction into consideration to address this problem. Customer satisfaction affects the profit of cloud service providers in two ways. On one hand, the cloud configuration affects the quality of service which is an important factor affecting customer satisfaction. On the other hand, the customer satisfaction affects the request arrival rate of a cloud service provider. However, few existing works take customer satisfaction into consideration in solving profit maximization problem, or the existing works considering customer satisfaction do not give a proper formalized definition for it. Hence, we firstly refer to the definition of customer satisfaction in economics and develop a formula for measuring customer satisfaction in cloud computing. And then, an analysis is given in detail on how the customer satisfaction affects the profit. Lastly, taking into consideration customer satisfaction, service-level agreement, renting price, energy consumption and so forth, a profit maximization problem is formulated and solved to get the optimal configuration such that the profit is maximized.

Index Terms—cloud computing; customer satisfaction; multiserver system; profit maximization; PoS; QoS; service-level agreement;

1 Introduction

Cloud computing is the delivery of resources and computing as a service rather than a product over the Internet, such that accesses to shared hardware, software, databases, information, and all resources are provided to consumers on-demand [1]. Customers use and pay for services on-demand without considering the upfront infrastructure costs and the subsequent maintenance cost [2]. Due to such advantages, cloud computing is becoming more and more popular and has received considerable attention recently. Nowadays, there have been many cloud service providers, such as Amazon EC2 [3], Microsoft Azure [4], Saleforce.com [5], and so forth.

As a kind of new IT commercial model, profit is an important concern of cloud service providers. As shown in Fig. 1, the cloud service providers rent resources from infrastructure providers to configure the service platforms and provide paid services to customers to make profits. For cloud service providers, how to configure their cloud service platforms to obtain the maximal profit becomes increasingly the focus that they pay attention to.

The optimal configuration problem with profit maximization of cloud service providers has been researched

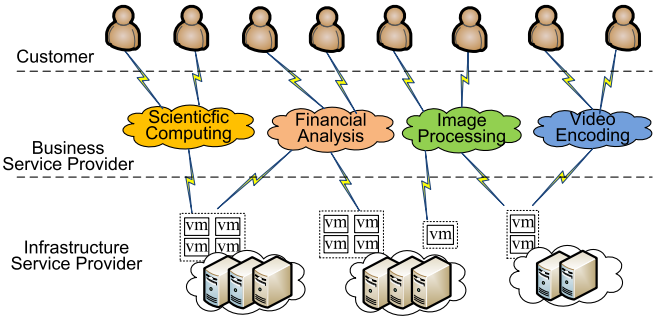


Fig. 1: The Three-Tier Cloud Structure

in our previous researches [2, 6] which assumed that the cloud service demand is known in advance and not affected by external factors. However, the request arrival rate of a service provider is affected by many factors in actual, and customer satisfaction is the most important factor. For example, customers could submit their tasks to a cloud computing platform or execute them on their local computing platforms. The customer behavior depends on if the cloud service is attractive enough to them. To configure a cloud service platform properly, the cloud service provider should know how customer satisfaction affects the service demands. Hence, considering customer satisfaction in profit optimization problem is necessary. However, few existing works take customer satisfaction into consideration in solving profit maximization problem, or the existing works considering customer satisfaction do not give a proper formalized definition for it. To address the problem, this paper adopts the thought in Business Administration, and firstly defines the *customer satisfaction level* of cloud computing.

- *Jing Mei is with the College of Mathematics and Computer Science, Hunan Normal University in Changsha, Hunan, China, 410006. E-mail: jingmei1988@163.com*
- *Kenli Li and Keqin Li are with the College of Information Science and Engineering, Hunan University, and National Supercomputing Center in Changsha, Hunan, China, 410082. E-mail: lkl@hnu.edu.cn, lik@newpaltz.edu.*
- *Keqin Li is also with the Department of Computer Science, State University of New York, New Paltz, New York 12561, USA.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

Based on the definition of customer satisfaction, we build a profit maximization model in which the effect of customer satisfaction on quality of service (QoS) and price of service (PoS) is considered. From an economic standpoint, two factors affecting customer satisfaction are QoS and PoS. The PoS is determined by cloud service providers. The QoS is determined by the service capacity of a cloud service provider which largely depends on its platform configuration. Under the given pricing strategy, the only way to improve the customer satisfaction level is to promote the QoS, which can be achieved by configuring cloud platform with higher service capacity. Doing so can affect a cloud service provider from two asides. On one hand, the higher customer satisfaction level leads to a higher market share, so the cloud service provider can gain more revenues. On the other hand, more resources are rented to improve the service capacity, which leads to the increase of costs. Hence, the ultimate solution of improving profit is to find an optimal cloud platform configuration scheme. In this paper, we build a customer-satisfaction-aware profit optimization model and propose a discrete hill climbing algorithm to find the numeric optimal cloud configuration for cloud service providers.

The contributions of this paper are listed as follows:

- Based on the definition of customer satisfaction level in economics, develop a calculation formula for measuring customer satisfaction in cloud;
- Analyze the interrelationship between customer satisfaction and profit, and build a profit optimization model considering customer satisfaction;
- Develop a discrete hill climbing algorithm to find the optimal cloud configuration such that the profit is maximized.

The rest of the paper is organized as follows. Section 2 reviews the related work on profit maximization in cloud computing. Section 3 gives a definition of customer satisfaction level and its calculation formula. Section 4 presents the cloud service system model and the service-level agreement adopted in this paper. In Section 5, the customer satisfaction of cloud service providers is calculated. Section 6 builds the profit optimization model and proposes a heuristic algorithm to find the optimal cloud configuration. Section 7 conducts a series of numerical calculations to analyze the changing trend of the customer satisfaction and the profit with varying cloud configuration. A group of comparisons are conducted to prove the superiority of our method. Finally, Section 8 concludes the works.

2 Related Work

In this section, we firstly review the literatures concerning customer satisfaction, and then the profit maximization problem in cloud computing.

To estimate the service demand of a service provider, it is critical to measure its customer satisfaction. In business management, there have been many specialists who focus on the researches of the definition of customer satisfaction [7, 8, 9, 10, 11]. The concept of customer satisfaction is firstly proposed by Cardozo [7] in 1965 and he believed that high customer satisfaction produces purchase behavior again. After that, many different definitions are proposed for customer satisfaction. Howard and Sheth [8] considered customer satisfaction as the psychological states of a customer when evaluating the reasonability of pay and gain. Churchill and Surprenant [9] considered customer satisfaction as the comparison results between the payment to buy a product or service and the benefit using this product or service. Tes and Wilton [10] defined customer satisfaction as evaluation of the difference between prior expectation and cognitive performance. Parasuraman *et al.* [11] believed that customer satisfaction is a function of QoS and PoS. Although these definitions are described differently, their ideas are consistent with that of discrepancy theory [12, 13], that is, in any case, customer satisfaction is determined by the difference between prior expectation and actual cognitive afterwards.

In recent years, cloud computing has become a booming service industry. How to increase profit is an important issue for cloud service providers. Many works have been done to research this issue [2, 14, 15, 16, 17, 18, 19]. There are some researches focusing on the profit maximization problem of the service providers. Chaisiri *et al.* [18] took into consideration the uncertainty of the customers demand, and proposed a stochastic programming model with two-stage recourse to solve the profit maximization problem for the service providers. Cao *et al.* [2] proposed an optimal multiserver configuration strategy. Through the optimal strategy, the optimal configuration of multiserver system, i.e., the server size and the server speed, can be determined such that the profit of a multiserver system is maximized. Some papers consider the profit problem under different cloud computing environments. For example, Liu *et al.* [19] considered a cloud service provider operating geographically distributed data centers in a multi-electricity-market environment, and proposed an energy-efficient, profit- and cost-aware request dispatching and resource allocation algorithm to maximize a service providers net profit. In above works, they did not take customer satisfaction into consideration.

There are some works in cloud computing which consider customer satisfaction [20, 21, 22, 23, 24, 25, 26, 27, 28]. Chen *et al.* [20] adopted utility theory leveraged from economics and developed an utility model for measuring customer satisfaction in cloud. In the utility model, consumer satisfaction is relevant to two factors: service price and response time. They assumed that consumer satisfaction is decreased with higher service price and longer response time. In [21], the user satisfaction is calculated as the ratio of the actual QoS level and the expected QoS level. Wu *et al.* [22] proposed an admission control and scheduling algorithms for SaaS providers to maximize profit by minimizing cost and improve customer satisfaction level. However, they did not give a specific formula to measure customer satisfaction level. Chao *et al.* [24] proposed a customer satisfaction-aware algorithm based on the Ant-Colony Optimization (AMP) for geo-distributed datacenters. In this paper, the customer satisfaction model is same as that used in [20]. In [26], the authors defined the users' satisfaction as the extent to which the user's resource requirements have been met, and it is calculated as the ratio of the actual consumption and the expectation resources. In [27], Unuvar *et al.* proposed a predictive approach to select an optimum cloud availability zone that maximizes user satisfaction. However,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

the user satisfaction here is defined as how much the requirements specified in a request are satisfied. Morshedlou and Meybodi [28] defined the users' satisfaction level based on expected value of user's utility that an user attaches to a certain monetary amount. However, the existing formulas measuring customer satisfaction of cloud computing cannot properly reflect the definition of customer satisfaction, and they did not take into account user's psychological differences.

To address this problem, we use the definition of customer satisfaction leveraged from economics and develop a formula to measure customer satisfaction in cloud. And then, how cloud configuration affects customer satisfaction and how customer satisfaction affects the profit of cloud service providers are analyzed. Based on these works, a profit maximization problem considering customer satisfaction is formulated and solved such that the optimal configuration is obtained.

3 The Definition of Customer Satisfaction Level

Customer satisfaction is an important factor that should be considered in a service market, i.e., cloud computing, which is a measure of how products and services supplied by a company meet or surpass customer expectation [9, 10], and it directly affects the number of customers of a company, and the profit consequently. In general, the overall customer satisfaction level of a company is an accumulation of the satisfaction values of all customers. In the following, we first give the satisfaction formula of each customer, and then the overall customer satisfaction of a company.

3.1 Satisfaction of Single Customer

The QoS and PoS are two main factors affecting the service evaluation of each customer. Hence, we define the satisfaction of a customer $\mathcal{S}_{one}$ as the product of QoS satisfaction and PoS satisfaction as follows:

$$\mathcal{S}_{one} = \mathcal{S}_{QoS}\mathcal{S}_{PoS}, \tag{1}$$

where $\mathcal{S}_{QoS}$ and $\mathcal{S}_{PoS}$ are the QoS satisfaction and PoS satisfaction, respectively.

QoS satisfaction: From a psychological point, QoS is a subjective concept which is the result of the comparison that customers make between their expectations about a service and their perceptions of the way the service has been performed [11, 29, 30, 31]. The expectations are not generated out of thin air but based on the established price. For example, if the PoS of a provider is high, which implies that its QoS would be better than those providers with a lower price, hence, the customers' expectations of performance would be higher. Under a given price, if the perceptions of performance surpass the expectations, the QoS is considered as high, and vice verse. High QoS makes a high QoS satisfaction, and with the decreasing of QoS, the QoS satisfaction is dropping continuously. Hence, the true factor which affects QoS satisfaction is the discrepancy between the perception performance and the expectation performance. According to the discrepancy theory in economics, we formulate the QoS satisfaction of a customer as

$$\mathcal{S}_{QoS} = \begin{cases} 1, & \textbf{if } \mathcal{P}_{per} \geq \mathcal{P}_{exp}; \\ e^{-|(\mathcal{P}_{per}-\mathcal{P}_{exp})/\mathcal{P}_{exp}|}, & \\ & \textbf{if } \mathcal{P}_{per} < \mathcal{P}_{exp}, \end{cases} \tag{2}$$

where $\mathcal{P}_{per}$ and $\mathcal{P}_{exp}$ present the perception performance and the expectation performance, respectively. This equation is proposed based on the understanding of Kano model which is proposed by Kano *et al.* in [32] and it is a theory of product development and customer satisfaction.

PoS satisfaction: Similarly, the PoS satisfaction can be formulated as the comparison between the predefined price and the actual price, which is defined as

$$\mathcal{S}_{PoS} = e^{(\mathcal{C}_{pre}-\mathcal{C}_{act})/\mathcal{C}_{pre}}, \tag{3}$$

where $\mathcal{C}_{pre}$ and $\mathcal{C}_{act}$ present the predefined price and the actual price, respectively. In general, the PoS of a service provider is pre-made. Before a customer submits the requests, the PoS is known which can be considered as the expected price. If the actual PoS is equal to the expected price, we consider the default satisfaction in terms of price to be 1, that means, the price has no effect on the total satisfaction. If the actual PoS is higher than the expected price, the PoS satisfaction is less than 1 and decreases with the increasing PoS. On the contrary, if the actual PoS is lower than the expected price, the customer can be delighted by the low price, hence the PoS satisfaction is greater than 1 and increases with the decreasing PoS.

3.2 Overall Customer Satisfaction

According Eq. (1), it is easy to estimate the satisfaction of each customer. However, what really matters is the overall customer satisfaction of a service provider (denoted by $\mathcal{S}$), which is the expectation of satisfaction of all customers as Eq. (4):

$$\mathcal{S} = \overline{\mathcal{S}_{one}}. \tag{4}$$

Since the objective of this paper is to research the profit optimization of cloud service providers, we should study how to measure the customer satisfaction of a cloud service provider and how the customer satisfaction affects the its profit, which are analyzed in the following.

4 Preliminary Knowledge

Before analyzing the customer satisfaction of a cloud service provider, we present the service model first. Besides, Service-Level Agreement (SLA) is also introduced, which is a negotiation about the charge and the QoS between cloud service providers and customers.

4.1 The Cloud Service Model

The cloud service system is a multiserver system shown in Fig. 2 which can be modeled as an M/M/m queuing model. Similar models are used in many researches on cloud computing such as [2, 6, 33].

In the M/M/m model, m is the number of servers, and all servers run at an identical speed s (measured by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

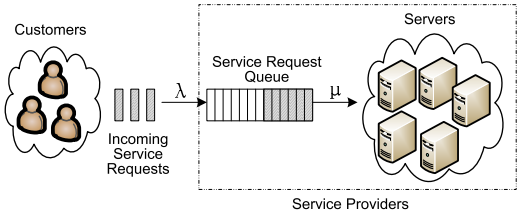


Fig. 2: The M/M/m Queuing Model

the number of instructions that can be executed in one unit of time). Assume that the interarrival times of service requests are independent and identically distributed (i.i.d.) exponential random variables, in other words, the arrival requests follow a Poisson process with arrival rate λ [2]. The execution requirements of the tasks (measured by the number of instructions to be executed) are i.i.d. exponential random variables r with mean $\overline{r}$. Since the server execution speed is s, the service times of the requests are also i.i.d. exponential random variables $x = r/s$ with mean $\overline{x} = \overline{r}/s$. Hence, the average service rate, i.e., the average number of service requests that can be completed by a server with speed s in one unit of time, is $\mu = 1/\overline{x} = s/\overline{r}$.

Assume that the number of virtual machines in a server is fixed and cannot be changed during the runtime. Each arriving request enters the multiserver system and waits in a queue with infinite capacity when all the servers are busy. The first-come-first-served (FCFS) queuing strategy is adopted. Let π_k denote the probability that there are k service requests (waiting or being processed) in the M/M/m queuing system. We have

$$\pi_k = \begin{cases} \pi_0 \frac{(m\rho)^k}{k!}, & k \le m; \\ \pi_0 \frac{m^m \rho^k}{k!}, & k > m, \end{cases}$$

where

$$\pi_0 = \left(\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right)^{-1},$$

and ρ is the server utilization which is calculated as $\rho = \lambda/m\mu = \lambda\overline{x}/m = \lambda/m \cdot \overline{r}/s$.

If all servers are busy when a newly service request is submitted, it must wait and the probability is

$$\Pi_q = \sum_{k=m}^{\infty} \pi_k = \frac{\pi_m}{1-\rho}.$$

Let W denote the waiting time of a newly arrived service request to the multiserver M/M/m system. The *probability distribution function (pdf)* of the waiting time W is

$$f_W(t) = (1 - \Pi_q)u(t) + m\mu\pi_m e^{-(1-\rho)m\mu t}, \tag{5}$$

where $u(t)$ is a unit impulse function defined as:

$$u_z(t) = \begin{cases} z, & 0 \le t \le \frac{1}{z}; \\ 0, & t > \frac{1}{z}, \end{cases}$$

and

$$u(t) = \lim_{z \to \infty} u_z(t).$$

The function $u_z(t)$ has the following properties, i.e.,

$$\int_0^{\infty} u_z(t)dt = 1$$

and

$$\int_0^{\infty} t u_z(t)dt = z \int_0^{1/z} t dt = \frac{1}{2z}.$$

4.2 The Service-Level Agreement

In general, the QoS is affected by many factors such as the service time, the failure rate and so forth. However, in this paper, we measure the QoS of a request by its response time for two reasons. First, the service time is easily measured. Second, it gives customers an intuitive feeling of QoS. For customers, they do not care how failures are managed when failures occur. They only care whether the task can be completed successfully and how long it takes.

The response times of requests are different from each other due to the changing system workload and limited service capacity, which leads to different QoS and QoS satisfaction. In general, each customer has a tolerable response time which is related to the execution requirement of its requests. We denote the tolerable response time of a request with execution requirement r by cr/s_0, where s_0 is be baseline speed of a server and c is a constant coefficient. If the response time of a request exceeds the tolerable value, the customer feels dissatisfaction about the service, which leads to the degrade of the overall customer satisfaction of the service provider.

To protect the interests of customers and maintain the customer satisfaction, there is always a service-level agreement (SLA) between a service provider and customers in which the QoS and the corresponding charge are stipulated. In this paper, we adopt a similar SLA with [2] which defines the service charge for a service request with execution requirement r and response time T to be

$$C(r,T) = \begin{cases} ar, & \textbf{if } 0 \le T \le \frac{c}{s_0} r; \\ ar - \ell\left(T - \frac{c}{s_0} r\right), & \\ & \textbf{if } \frac{c}{s_0} r < T \le \left(\frac{a}{\ell} + \frac{c}{s_0}\right) r; \\ 0, & \textbf{if } T > \left(\frac{a}{\ell} + \frac{c}{s_0}\right) r, \end{cases} \tag{6}$$

where a is the service charge per unit amount of service and ℓ is a coefficient representing the compensation strength due to the low QoS.

This SLA stipulates how to compensate the customers when the QoS is low. In this case, a common approach adopted by service providers is reducing the charge as a compensation of low QoS to maintain the customer satisfaction. If the response time T of serving a request is not longer than $(c/s_0)r$, then the service request is processed with high QoS and the customer is charged ar. If the response time T is longer than $(c/s_0)r$ but not longer than $(a/\ell + c/s_0)r$, then the service request is served with low quality and the charge to a customer decreases linearly as T increases. If the response time T is longer than $(a/\ell + c/s_0)r$, the service is free [2].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

5 Customer Satisfaction of Cloud Service Providers

In the following, the measurement of customer satisfaction of a cloud service provider is introduced based on Eq. (4).

The waiting time W of a service request is $W = T - r/s$, where s is the actual speed of a server which can be decided by a service provider and r/s is the actual execution time under speed s. Since the distribution function of waiting time W of service requests is known, it is better to rewrite Eq. (6) in terms r and waiting time W instead of response time T. The rewritten charge function is

$$C(r,W) = \begin{cases} ar, & \textbf{if } 0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r; \\ (a + \frac{c\ell}{s_0} - \frac{\ell}{s})r - \ell W, & \\ \qquad \textbf{if } (\frac{c}{s_0} - \frac{1}{s})r < W \le (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r; \\ 0, & \textbf{if } W > (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r. \end{cases} \tag{7}$$

If we know the QoS of a request and its corresponding charge, it is easy to estimate its service satisfaction.

5.1 The Calculation of Single Customer Satisfaction

The expectation response time of a request with requirements r is cr/s_0, that is, its expectation waiting time is $(cr/s_0 - 1/s)$. Assume that its actual waiting time is W and the actual price is x (x is the price per unit amount of service), then its QoS satisfaction is formulated as

$$\mathcal{S}_{QoS}(r,W) = \begin{cases} e^{\frac{(c/s_0 - 1/s)r - W}{(c/s_0 - 1/s)r}}, \textbf{ if } W > (\frac{c}{s_0} - \frac{1}{s})r; \\ 1, \textbf{ if } 0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r, \end{cases} \tag{8}$$

and its PoS satisfaction is formulated as

$$\mathcal{S}_{PoS}(r,x) = \begin{cases} 1, & \textbf{if } x = a; \\ e^{\frac{\ell}{a}(\frac{1}{s} + \frac{W}{r} - \frac{c}{s_0})}, & \\ \qquad \textbf{if } x = (a + \frac{c\ell}{s_0} - \frac{\ell}{s}) - \ell W/r; \\ e, & \textbf{if } x = 0. \end{cases} \tag{9}$$

Because the actual price x is determined by the waiting time W which is $x = C(r,W)/r$, Eq. (9) can be rewritten by replacing the independent variable x with W:

$$\mathcal{S}_{PoS}(r,W) = \begin{cases} 1, & \textbf{if } 0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r; \\ e^{\frac{\ell}{a}(\frac{1}{s} + \frac{W}{r} - \frac{c}{s_0})}, & \\ \qquad \textbf{if } (\frac{c}{s_0} - \frac{1}{s})r < W \le (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r; \\ e, & \textbf{if } W > (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r. \end{cases} \tag{10}$$

Substituting Eqs. (8) and (10) into Eq. (1), we can get the service satisfaction of a request with r and W as

$$\mathcal{S}_{one}(r,W) = \begin{cases} 1, & \textbf{if } 0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r; \\ e^{1 + \frac{\ell}{a}(\frac{1}{s} - \frac{c}{s_0}) + (\frac{\ell}{a} - \frac{1}{c/s_0 - 1/s})\frac{W}{r}}, & \\ \qquad \textbf{if } (\frac{c}{s_0} - \frac{1}{s})r < W \le (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r; \\ e^{2 - \frac{W}{(c/s_0 - 1/s)r}}, & \textbf{if } W > (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r. \end{cases} \tag{11}$$

In the given charge function of Eq. (6), ℓ is a constant representing the compensation degree when the QoS cannot meet the expectation and it is determined by service providers. How to set ℓ is a problem for service providers because it has a direct relation with their profit. Obviously, a greater ℓ can improve the customer satisfaction more efficiently, but it also leads to lower revenues. In this paper, we assume that the range of customer satisfaction is between 0 and 1. Then, the ℓ value should be selected properly such that each customer's satisfaction is not greater than 1, and a proper range of ℓ is given in Theorem 5.1.

Theorem 5.1. *Let the maximal satisfaction of a customer be 1. ℓ should be less than or equal to $a/(\frac{c}{s_0} - \frac{1}{s})$.*

Proof. For a request with service requirement r and waiting time W, its customer satisfaction can be analyzed in three situations according to Eq. (11):

- If $0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r$, the customer satisfaction is always 1.
- If $(\frac{c}{s_0} - \frac{1}{s})r < W \le (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r$, the customer satisfaction is $\mathcal{S}_{one}\ e^{1 + \frac{\ell}{a}(\frac{1}{s} - \frac{c}{s_0}) + (\frac{\ell}{a} - \frac{1}{c/s_0 - 1/s})\frac{W}{r}}$ which is a monotone function of W since $(\frac{\ell}{a} - \frac{1}{c/s_0 - 1/s})$ is a constant. To guarantee the range of satisfaction be [0, 1], the satisfaction values at $W_1 = (\frac{c}{s_0} - \frac{1}{s})r$ and $W_2 = (\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r$ are not greater than 1, and the corresponding inequality equation is

$$\begin{aligned} &\mathcal{S}_{one}(r, W_1) \\ &= e^{1 + \frac{\ell}{a}(\frac{1}{s} - \frac{c}{s_0}) + (\frac{\ell}{a} - \frac{1}{c/s_0 - 1/s})\frac{(c/s_0 - 1/s)r}{r}} \\ &= 1. \end{aligned} \tag{12}$$

$$\begin{aligned} &\mathcal{S}_{one}(r, W_2) \\ &= e^{1 + \frac{\ell}{a}(\frac{1}{s} - \frac{c}{s_0}) + (\frac{\ell}{a} - \frac{1}{c/s_0 - 1/s})\frac{(a/\ell + c/s_0 - 1/s)r}{r}} \\ &= e^{1 - \frac{a/\ell}{c/s_0 - 1/s}} \\ &\le 1. \end{aligned} \tag{13}$$

  Solving Eq. (13), we can get

$$\ell \le \frac{a}{c/s_0 - 1/s}.$$

- When W is longer than $(\frac{a}{\ell} + \frac{c}{s_0} - \frac{1}{s})r$,

$$\begin{aligned} \mathcal{S}_{one}(r,W) &= e^{2 - \frac{W}{(c/s_0 - 1/s)r}} \\ &\le e^{1 - \frac{a/\ell}{c/s_0 - 1/s}} \\ &\le 1. \end{aligned} \tag{14}$$

  Similarly, the solution is

$$\ell \le \frac{a}{c/s_0 - 1/s}.$$

To sum up, ℓ is not greater than $a/(\frac{c}{s_0} - \frac{1}{s})$. This completes the proof of the theorem. □

In this paper, we set $\ell = a/(\frac{c}{s_0} - \frac{1}{s})$ to maximize the customer satisfaction, so Eq. (11) is simplified as

$$\mathcal{S}_{one}(r,W) = \begin{cases} 1, & \textbf{if } 0 \le W \le (\frac{c}{s_0} - \frac{1}{s})r; \\ 1, & \textbf{if } (\frac{c}{s_0} - \frac{1}{s})r < W \le 2(\frac{c}{s_0} - \frac{1}{s})r; \\ e^{2 - \frac{W}{(c/s_0 - 1/s)r}}, & \textbf{if } W > 2(\frac{c}{s_0} - \frac{1}{s})r. \end{cases}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

and Eq. (7) is simplified as

$$C(r,W)=\begin{cases} ar, & \textbf{if } 0\le W\le (\frac{c}{s_0}-\frac{1}{s})r;\\ 2ar-aW/(\frac{c}{s_0}-\frac{1}{s}), & \\ & \textbf{if } (\frac{c}{s_0}-\frac{1}{s})r<W\le 2(\frac{c}{s_0}-\frac{1}{s})r;\\ 0, & \textbf{if } W>2(\frac{c}{s_0}-\frac{1}{s})r. \end{cases}$$

5.2 The Overall Customer Satisfaction

Till now, the satisfaction of single customer has been given. The overall customer satisfaction $\mathcal{S}$ of a service provider is the expectation of satisfaction of all customers. The following theorem gives the customer satisfaction of a service provider.

Theorem 5.2. *The overall customer satisfaction of a service provider is*

$$\mathcal{S}=1-\frac{\Pi_q}{\bar{r}}\int_0^{\infty}\frac{e^{-((1-\rho)m\mu\cdot 2(c/s_0-1/s)+1/\bar{r})z}}{(1-\rho)m\mu(c/s_0-1/s)z+1}dz, \qquad (15)$$

where $\Pi_q=\pi_m/(1-\rho)$ *and* $\pi_m=\pi_0(m\rho)^m/m!$.

Proof. Since W is a random variable, $\mathcal{S}_{one}(r,W)$ is also a random variable because it is a function of W for a fixed r. The customer satisfaction of a service provider which serves requests with the same execution requirement r is

$$\begin{aligned}
&\mathcal{S}(r)\\
&=\overline{\mathcal{S}_{one}(r,W)}\\
&=\int_{-\infty}^{\infty}f_W(t)\mathcal{S}_{one}(r,t)dt\\
&=\int_0^{2(\frac{c}{s_0}-\frac{1}{s})r}f_W(t)dt+\int_{2(\frac{c}{s_0}-\frac{1}{s})r}^{\infty}f_W(t)e^{2-\frac{t}{(c/s_0-1/s)r}}dt\\
&=\int_0^{2(\frac{c}{s_0}-\frac{1}{s})r}\big((1-\Pi_q)u(t)+m\mu\pi_m e^{-(1-\rho)m\mu t}\big)dt\\
&+\int_{2(\frac{c}{s_0}-\frac{1}{s})r}^{\infty}\big((1-\Pi_q)u(t)+m\mu\pi_m e^{-(1-\rho)m\mu t}\big)e^{2-\frac{t}{(c/s_0-1/s)r}}dt\\
&=1-\Pi_q+\frac{\pi_m}{1-\rho}\big(1-e^{-(1-\rho)m\mu\cdot 2(c/s_0-1/s)r}\big)\\
&+\frac{m\mu\pi_m(c/s_0-1/s)r}{(1-\rho)m\mu(c/s_0-1/s)r+1}e^{-(1-\rho)m\mu\cdot 2(c/s_0-1/s)r}\\
&=1-\frac{\Pi_q}{(1-\rho)m\mu(c/s_0-1/s)r+1}e^{-(1-\rho)m\mu\cdot 2(c/s_0-1/s)r}.
\end{aligned}$$

Since r is an exponential random variable and its pdf is $f_r(z)=\frac{1}{\bar{r}}e^{-z/\bar{r}}$, $\mathcal{S}(r)$ is also a random variable and its expectation is

$$\begin{aligned}
&\mathcal{S}\\
&=\overline{\mathcal{S}(r)}\\
&=\int_{-\infty}^{\infty}f_r(z)\mathcal{S}(z)dz\\
&=\int_0^{\infty}\frac{e^{-z/\bar{r}}}{\bar{r}}dz-\int_0^{\infty}\frac{e^{-z/\bar{r}}}{\bar{r}}\frac{\Pi_q e^{-(1-\rho)m\mu\cdot 2(c/s_0-1/s)z}}{(1-\rho)m\mu(c/s_0-1/s)z+1}dz\\
&=1-\frac{\Pi_q}{\bar{r}}\int_0^{\infty}\frac{e^{-((1-\rho)m\mu\cdot 2(c/s_0-1/s)+1/\bar{r})z}}{(1-\rho)m\mu(c/s_0-1/s)z+1}dz.
\end{aligned}$$

The theorem is proved. □

It is easy to know that the overall customer satisfaction $\mathcal{S}$ is related with λ, $\bar{r}$, m and s, and its range is [0,1]. Because the analytical solutions of $\mathcal{S}$ cannot be solved, its numerical solutions are adopted in the rest calculations.

6 The Profit Optimization Problem

In this Section, how the customer satisfaction of a service provider affects its profit is first analyzed. And then the profit optimization model is build to find the optimal configuration of cloud service providers.

6.1 Customer Satisfaction Aware Arrival Rate

In a market economy, the customer satisfaction of a service provider affects its market share. Assume that the total market demand is λ_{max}, the market share $\mathcal{M}_{\mathcal{S}}$ of a service provider is the ratio of the actual task arrival rate λ and λ_{max} which can be formulated as

$$\mathcal{M}_{\mathcal{S}}=\lambda/\lambda_{max}.$$

In general, a higher customer satisfaction would lead to a larger market share, but the growth trends are different in different situations. In this paper, we assume that the market share $\mathcal{M}_{\mathcal{S}}$ of a service provider is linearly increasing with its customer satisfaction which is denoted as

$$\mathcal{M}_{\mathcal{S}}=\mathcal{S}.$$

Combining above two equations, we can get the relationship between the actual task arrival rate λ and the custom satisfaction $\mathcal{M}_{\mathcal{S}}$ as

$$\lambda=\mathcal{S}\lambda_{max}. \qquad (16)$$

Substituting Eq. (15) into Eq. (16), we can get the task arrival rate of a service provider in steady sate by solving Eq. (16). Eq. (16) is so complicated that we cannot find a closed-form solution of λ. However, we can obtain a numerical solution of λ for it.

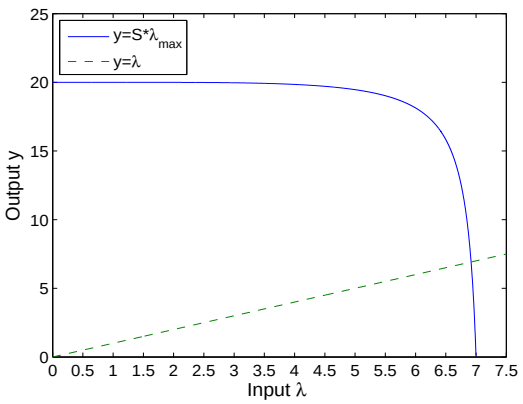


Fig. 3: The changing trend of λ.

Fig. 3 gives the graph of $y=\mathcal{S}\lambda_{max}$ and $y=\lambda$. From Fig. 3 it is easy to know that function $y=\mathcal{S}\lambda_{max}$ is monotonic decreasing and $y=\lambda$ is monotonic increasing, so

$$D(\lambda)=\mathcal{S}\lambda_{max}-\lambda \qquad (17)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

is a decreasing function of λ. Hence, we can adopt the standard bisection method to find a numerical solution of λ and the process is given as Algorithm 1. In Algorithm 1, the input is arbitrary multiserver configuration (m, s), and the output is the actual arrival rate $\lambda_{m,s}$ of the service provider with configuration (m, s).

Algorithm 1 Actual Arrival Rate $\lambda_{m,s}$

Input: multiserver configuration m and s;
Output: the actual task arrival rate, $\lambda_{m,s}$;
1: find the monotone interval $[\lambda_l, \lambda_u]$ of $D(\lambda)$ such that $D(\lambda_l) > 0$ and $D(\lambda_u) < 0$;
2: **while** $D(\lambda_l) - D(\lambda_u) > \varepsilon$ **do**
3: $\lambda_{mid} \leftarrow (\lambda_l + \lambda_u)/2$;
4: **if** $G(\lambda_{mid}) < 0$ **then**
5: $\lambda_u \leftarrow \lambda_{mid}$;
6: **else**
7: $\lambda_l \leftarrow \lambda_{mid}$;
8: break;
9: **end if**
10: calculate $D(\lambda_l)$ and $D(\lambda_u)$ using Eq. (17);
11: **end while**
12: $\lambda_{mid} \leftarrow (\lambda_l + \lambda_u)/2$;
13: $\lambda_{m,s} \leftarrow \lambda_{mid}$;

6.2 The Profit Model

From the service providers' respective, the profit is mainly determined by the cost and the revenue.

6.2.1 The Cost Model

The cost of a service provider is mainly used to pay the rent and the electricity fee. A service provider rents servers from an infrastructure provider and pays the corresponding rent. The rent is determined by the number of rented servers and the rental price per server per unit of time. Assume that the rental price of one server per unit of time is β, and m servers are rented. The rent per unit of time is calculated as $E_{rent} = \beta m$.

Energy consumption is another major part of the cost paid by the service providers. In this paper, we focus on the compute-intensive service which consumes CPU resource mainly, hence, other energy consumption is assumed to be neglectable. Generally, the power consumption in digital CMOS circuits can be modeled as $P = P_d + P^*$, where P_d is dynamic power consumption and P^* is the power consumption when a server is idle [34]. In this paper, we set $P_d = \xi s^\alpha$ where $\alpha = 2.0$ and $\xi = 9.4192$, and the energy consumption formula is widely used. In the M/M/m queuing system, the server utilization is ρ, then the average amount of dynamic energy consumption of an m-server system with speed s per unit of time is $m\rho\xi s^\alpha$. Let the cost of energy be γ per Watt. The total cost of energy consumption of the m-server system in one unit of time is $E_{energy} = \gamma m(\rho\xi s^\alpha + P^*)$.

Based on the analysis above, the cost of a service provider is a sum of the rent and the energy consumption cost as follows:

$$E = \beta m + \gamma m(\rho\xi s^\alpha + P^*).$$

6.2.2 The Revenue Model

To calculate the revenue of a service provider, we should know the expected charge to a service request, which is given in Theorem 6.1.

Theorem 6.1. *The expected charge to a service request is*

$$C = a\bar{r}\left(1 - \frac{\Pi_q}{(2(ms-\lambda\bar{r})(\frac{c}{s_0}-\frac{1}{s})+1)((ms-\lambda\bar{r})(\frac{c}{s_0}-\frac{1}{s})+1)}\right). \tag{18}$$

Proof. The proof is similar to that of Theorem 5.2. First, the expected charge to a request with execution requirement r is

$$\begin{aligned}
&C(r) = \overline{C(r,W)} \\
&= \int_{-\infty}^{\infty} f_W(t)C(r,t)dt \\
&= \int_0^{(\frac{c}{s_0}-\frac{1}{s})r} f_W(t)ardt + \int_{(\frac{c}{s_0}-\frac{1}{s})r}^{2(\frac{c}{s_0}-\frac{1}{s})r} f_W(t)\Big(2ar - \frac{at}{c/s_0 - 1/s}\Big)dt \\
&= ar(1-\Pi_q) + \int_0^{(\frac{c}{s_0}-\frac{1}{s})r} m\mu\pi_m e^{-(1-\rho)m\mu t}ardt \\
&+ \int_{(\frac{c}{s_0}-\frac{1}{s})r}^{2(\frac{c}{s_0}-\frac{1}{s})r} m\mu\pi_m e^{-(1-\rho)m\mu t}\Big(2ar - \frac{at}{c/s_0-1/s}\Big)dt.
\end{aligned}$$

Since

$$\int e^{-ax}dx = \frac{1}{-a}e^{-ax}$$

and

$$\int xe^{-ax}dx = \frac{1}{-a}(1+ax)e^{-ax},$$

we get

$$\begin{aligned}
&C(r) = \\
&= ar(1-\Pi_q) - \frac{\pi_m ar}{1-\rho}e^{-(1-\rho)m\mu t}\Big|_0^{(\frac{c}{s_0}-\frac{1}{s})r} \\
&- \frac{2\pi_m ar}{1-\rho}e^{-(1-\rho)m\mu t}\Big|_{(\frac{c}{s_0}-\frac{1}{s})r}^{2(\frac{c}{s_0}-\frac{1}{s})r} \\
&+ \frac{\pi_m a}{(c/s_0-1/s)(1-\rho)}\Big(t + \frac{1}{(1-\rho)m\mu}\Big)e^{-(1-\rho)m\mu t}\Big|_{(\frac{c}{s_0}-\frac{1}{s})r}^{2(\frac{c}{s_0}-\frac{1}{s})r} \\
&= ar - \Pi_q are^{-(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})r} \\
&- 2\Pi_q ar\big(e^{-(1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})r} - e^{-(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})r}\big) \\
&+ \frac{\Pi_q a}{c/s_0-1/s}\Big(2\Big(\frac{c}{s_0}-\frac{1}{s}\Big)r + \frac{1}{(1-\rho)m\mu}\Big)e^{-(1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})r} \\
&- \frac{\Pi_q a}{c/s_0-1/s}\Big(\Big(\frac{c}{s_0}-\frac{1}{s}\Big)r + \frac{1}{(1-\rho)m\mu}\Big)e^{-(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})r} \\
&= ar + \frac{\Pi_q a}{(c/s_0-1/s)(1-\rho)m\mu} \\
&\big(e^{-(1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})r} - e^{-(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})r}\big).
\end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

Then, the expected charge to a service request is

$$\begin{aligned}
C &= \overline{C(r)} \\
&= \int_0^\infty f_r(z)C(z)dz \\
&= \int_0^\infty \frac{1}{\overline{r}}e^{-z/\overline{r}}\Bigg(az+\frac{\Pi_q a}{(c/s_0-1/s)(1-\rho)m\mu} \\
&\left(e^{-(1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})z}-e^{-(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})z}\right)\Bigg)dz \\
&= \frac{1}{\overline{r}}\Bigg(a\int_0^\infty ze^{-z/\overline{r}}dz \\
&+\frac{\Pi_q a}{(c/s_0-1/s)(1-\rho)m\mu}\Big(\int_0^\infty e^{-((1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})+1/\overline{r})z}dz \\
&-\int_0^\infty e^{-((1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})+1/\overline{r})z}dz\Big)\Bigg).
\end{aligned}$$

Since

$$\int_0^\infty e^{-az}dz = \frac{1}{-a}e^{-az}\Big|_0^\infty = \frac{1}{a}$$

and

$$\int_0^\infty ze^{-az}dz = \frac{1}{-a}\Big(z+\frac{1}{a}\Big)e^{-az}\Big|_0^\infty = \frac{1}{a^2},$$

we get

$$\begin{aligned}
C &= a\overline{r}+\frac{\Pi_q a}{(c/s_0-1/s)(1-\rho)m\mu}\Bigg(\frac{1}{(1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})\overline{r}+1} \\
&-\frac{1}{(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})\overline{r}+1}\Bigg) \\
&= a\overline{r}+\frac{\Pi_q a}{(c/s_0-1/s)(1-\rho)m\mu} \\
&\left(\frac{(1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})\overline{r}}{((1-\rho)2m\mu(\frac{c}{s_0}-\frac{1}{s})\overline{r}+1)((1-\rho)m\mu(\frac{c}{s_0}-\frac{1}{s})\overline{r}+1)}\right) \\
&= a\overline{r}+\frac{\Pi_q a\overline{r}}{(2(ms-\lambda\overline{r})(c/s_0-1/s)+1)((ms-\lambda\overline{r})(c/s_0-1/s)+1)}.
\end{aligned}$$

The theorem is proved. □

Assume that the request arrival rate of a service provider is λ, then its revenue is

$$R = \lambda C.$$

6.3 Problem Description

In Section 6.1, the actual request arrival rate $\lambda_{m,s}$ of a service provider with server size m and server speed s has known, then the expected net profit of a service provider in one unit of time is

$$G(m,s)=\lambda_{m,s}C_{m,s}-(\beta m+\gamma m(\rho_{m,s}\xi s^\alpha+P^*)), \qquad (19)$$

where $C_{m,s}$ and $\rho_{m,s}$ is the expected charge serving a request and server utilization under the actual request demands.

From Eq. (19) we can see that the net profit is determined by the multiserver configuration scheme essentially. On one hand, the platform configuration directly affects the profit. On the other hand, the request requirement $\lambda_{m,s}$ is affected by customer satisfaction which largely depends on the service capacity of a cloud service provider. Hence, the platform configuration affects the profit indirectly. Generally speaking, configuring a cloud platform with more resources and faster speed can lead to a higher service capacity and a higher customer satisfaction. A higher customer satisfaction can attract more customers, hence lead to the increasing of the revenue. Whereas, a higher platform configuration also has a negative effect which is the costs is increasing correspondingly.

To maximize the profit of a service provider, an optimal configuration scheme should be decided by finding a solution $\langle m, s\rangle$ to the following optimization problem:

$$\max G(m,s). \qquad (20)$$

Fig. 4 gives the graph of $G(m,s)$ where $\lambda_{max}=20$, $s_0=1$, $\overline{r}=1$, $c=3$, $a=15$, $P^*=3$, $\alpha=2.0$, $\xi=9.4192$, $\beta=1.5$, and $\gamma=0.3$.

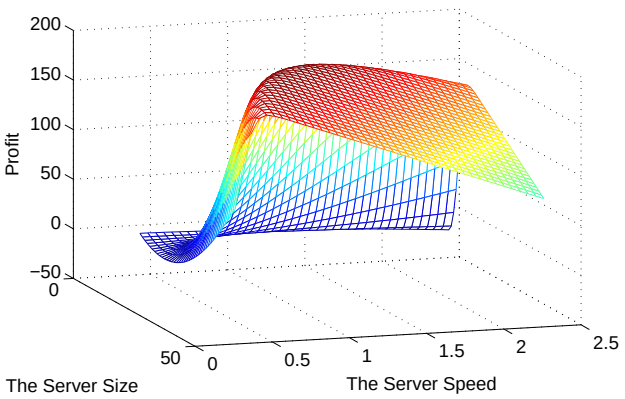


Fig. 4: The mesh of $G(m,s)$.

The figure shows that there must be an optimal point where the profit is maximized. However, we cannot give an analytical expression of profit in terms of m and s, so the analytical optimal solutions cannot be solved. To address this problem, we introduce a heuristic algorithm in next section to find a numerical optimal solution.

6.4 Algorithm for Optimal Multiserver Configuration

In this section, a heuristic algorithm is developed to find the optimal multiserver configuration such that the profit is maximized. According to the analysis above, it is known that under a fixed total market demand λ_{max}, the market share of a cloud service provider is different along with its different configuration, and the actual task arrival rate $\lambda_{m,s}$ at a given configuration can be calculated using Algorithm 1. Consequently, the net profit of a cloud service provider with different configuration is different, which can be calculated using Eq. (19).

To find the numerical optimal configuration, the search space should be discretized firstly. Under the discretized search space, the simplest way is brute force searching, e.g. calculating the profit of all possible configurations and selecting the optimal one. However, this brute force searching is not suitable due to high time complexity. To overcome this shortcoming, we propose a discrete hill climbing algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

The hill climbing is a numerical optimization technique which starts with an arbitrary solution to a problem and attempts to find a better solution by incrementally changing a single element of the solution [35].

For a traditional hill climbing algorithm, the sufficient condition of finding a global optimum is that the problem should be a convex problem. Fig. 5 shows the details of $G(m, s)$ in the range of server size [21,24] and server speed [0.9,1.3]. From Fig. 5, we can know that Eq. (19) is not a convex function because it has many extreme value points, so the search might stop in a local optimum when adopting traditional hill climbing algorithm. However, observing the mesh of $G(m, s)$ shown as Fig. 5, it is easy to know that the link line of all extreme value points shows a change trend of increasing firstly and then decreasing. Hence, to avoid the search stopping in a local optimum, when an extreme value point is found, the search goes on in the original direction until the next extreme value point is not greater than the current one.

The discrete hill climbing algorithm to solve this profit maximization problem is shown as Algorithm 2.

In Algorithm 2, the global optimum is found in the outer loop (lines 5-29), and the local optimum is found in the inner loop (lines 7-20). Firstly, the configuration with the maximal capacity is selected as the start search node (line 4), and it is considered as an initial global optimum (lines 4). The search starts from the start node. The inner loop compares the profit of the current node with its neighbour nodes, and lets the neighbour node with maximal profit be the next search node (lines 16-18). If none of the neighbour nodes can produce more profit than the current search node, the loop is stopped and an extreme value point is found (line 14). Comparing the current extreme value point with the current global optimum, if the current extreme point can generate more profit, it is updated as the new global optimum. At the same time, a new start search node is selected in the forward direction, and the search goes on to find the next extreme point (lines 22-25). If the current extreme point is not better than the former one, the outer while-loop is stopped and the global optimum point is found. The search process is illustrated as Fig. 6.

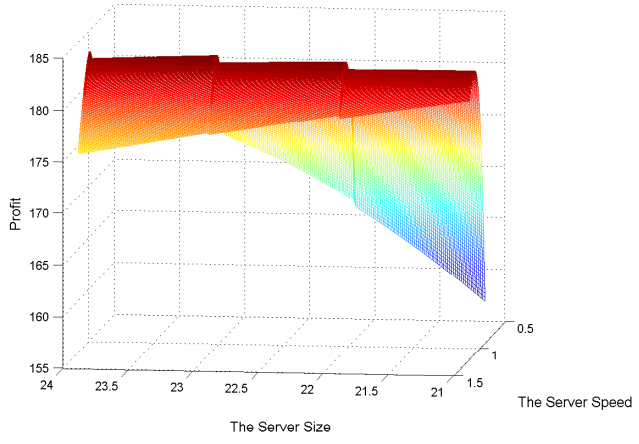


Fig. 5: The detail mesh of $G(m, s)$.

Algorithm 2 Optimal configuration

Input: λ_{max}, $\bar{r}$, $[M_{min}, M_{max}]$, $[S_{min}, S_{max}]$;
Output: optimal server size m_{opt}, optimal server speed s_{opt} and optimal profit Pro_{opt};

1: discretize $[M_{min}, M_{max}]$ and $[S_{min}, S_{max}]$;
2: set flag $\leftarrow 0$;
3: select (M_{max}, S_{max}) as start node (m, s);
4: $m_{opt} \leftarrow M_{max}$, $s_{opt} \leftarrow S_{max}$, $Pro_{opt} \leftarrow$ calculate $G(m, s)$ using Eq. (19);
5: **while** flag==0 **do**
6: initialize m_{curopt}, s_{curopt} and Pro_{curopt} as 0;
7: **while** true **do**
8: **for** each neighbour node (m, s) of current node **do**
9: $profit \leftarrow$ calculate $G(m, s)$ using Eq. (19);
10: **end for**
11: $(m_{tem}, s_{tem}) \leftarrow$ the neighbour node (m, s) with maximal profit;
12: $Pro_{tem} \leftarrow$ calculate $G(m_{tem}, s_{tem})$ using Eq. (19);
13: **if** $Pro_{tem} < Pro_{curopt}$ **then**
14: break;
15: **else**
16: $Pro_{curopt} \leftarrow Pro_{tem}$;
17: $m_{curopt} \leftarrow m_{tem}$;
18: $s_{curopt} \leftarrow s_{tem}$;
19: **end if**
20: **end while**
21: **if** $Pro_{curopt} > Pro_{opt}$ **then**
22: $Pro_{opt} \leftarrow Pro_{curopt}$;
23: $m_{opt} \leftarrow m_{curopt}$;
24: $Pro_{opt} \leftarrow Pro_{curopt}$;
25: select $(m_{opt} - 0.5, s_{opt})$ as new start node;
26: **else**
27: flag $\leftarrow 1$;
28: **end if**
29: **end while**

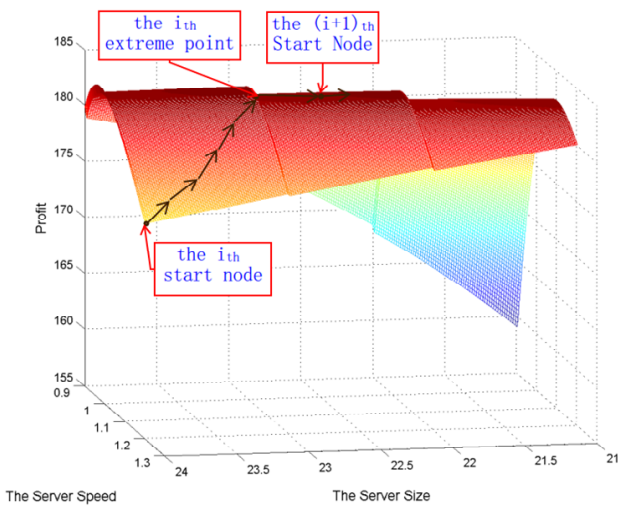


Fig. 6: The searching process.

7 Performance Analysis

In this section, a series of numerical calculations are conducted to observe the profit in different conditions, and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

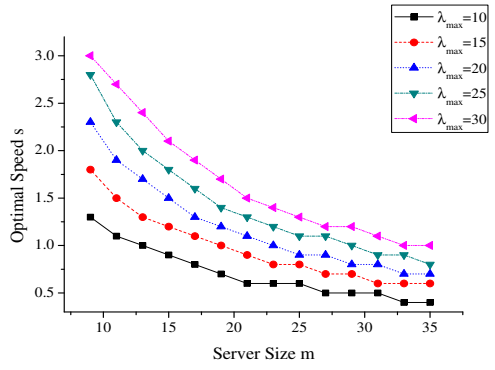
factors affecting the profit are analyzed.

7.1 Profit Optimization

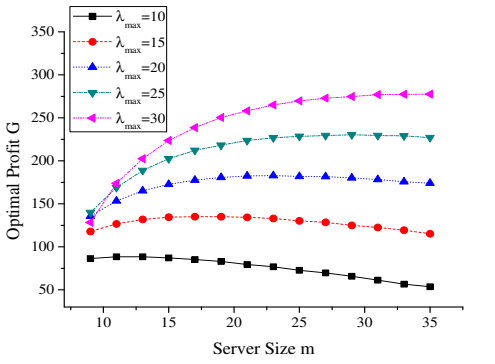
7.1.1 Optimal Speed

Given λ_{max}, s_0, $\bar{r}$, a, c, α, β, γ, ξ, P^*, δ, and m, our first group of numerical calculations are to find the optimal server speed s and the corresponding maximal profit G.

In Fig. 7(a) and Fig. 7(b), we demonstrate the optimal speed s and the corresponding profit G in one unit of time as a function of m and λ_{max}, respectively. Here, we assume that $s_0 = 1$ billion instructions per second, $a = 15$ cents per billion instructions, $c = 3$, $\alpha = 2.0$, $\beta = 1.5$ cents per second, $\gamma = 0.3$ cents per Watt×second, $\xi = 9.4192$, $P^* = 3$ Watts, and $\bar{r} = 1$ billion instructions. For λ_{max} =10, 15, 20, 25, 30, we show the changing trends of s and G for $5 \leq m \leq 20$.



(a) Optimal speed vs. Server size

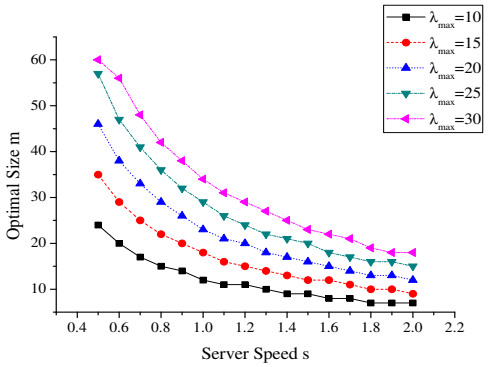


(b) Optimal profit vs. Server size

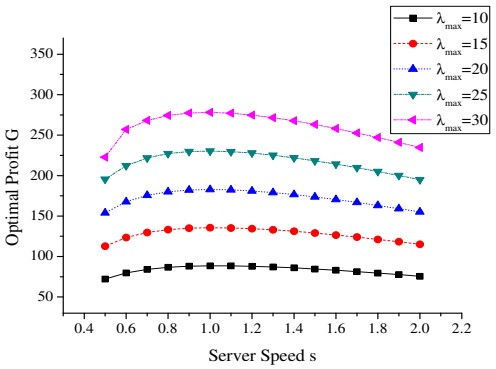
Fig. 7: Optimal speed and profit versus server size.

From Fig. 7(a), it is easy to see that the optimal speed s decreases with the increasing server size. That is because under the given market demand, the required service capacity is steady. Hence, when the number of servers configured in a cloud service platform increases, the server speed should drop to maintain the required capacity. Moreover, for a cloud platform with a fixed server size, when the market demand increases, the server should run faster to adapt this change. Correspondingly, Fig. 7(b) presents the changing trend of the optimal profit with the increasing server size and total market demand. It is obvious that greater market demand can bring higher profit for service providers. What's more, under a certain market demand, the server size also affect the profit, and there is an optimal choice of m such that the profit is maximized. For example, when the total market demand is 25, renting more servers can increase the profit when the server size is smaller than 15, but when the server size becomes greater further, the profit decreases. This is explained as follows. With the increase of server size, the server speed should decrease correspondingly to maintain a steady service capacity since the service requirement is limited. Because the energy cost is proportional to the square of the server speed, lowering the server speed can reduce the energy cost effectively when server is running fast. At the beginning, the server size is small and the server speed is very fast, so lowering the server speed and increasing the server size can reduce the overall costs because the reduced energy cost is greater than the extra cost of renting more servers. However, when the server speed decreases to a certain value, the increased renting cost of renting more servers starts surpassing the reduced cost of lowering the server speed. Hence, the profit starts decreasing. Hence, the server size is not the greater the better.

7.1.2 Optimal Size



(a) Optimal server size vs. Server speed



(b) Optimal profit vs. Server speed

Fig. 8: Optimal server size and profit versus server speed.

Given λ_{max}, ϱ, d, u, σ, α, β, γ, ξ, P^*, $\bar{r}$ and s, our second group of numerical calculations are to find the optimal server size m such that the net profit G is maximized.

In Fig. 8(a) and Fig. 8(b), we demonstrate the optimal server size m and the corresponding profit G in one unit of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

time as a function of s and λ_{max}, respectively. Here, we use the same parameters in Fig. 7. For λ_{max} =10, 15, 20, 25, 30, we show m and G for $0.1 \leq s \leq 2.0$.

Comparing Fig. 8 with Fig. 7, it is obvious that the changing trends of m and G in Fig. 7 are similar to that of s and G in Fig. 8. First, the optimal server size becomes smaller with the faster speed. Second, an optimal speed exists at a certain market demand. The reasons are explained as follows. If the server speed is too fast, the energy consumption cost would increase sharply, which leads to the decrease of the net profit. On the contrary, if the server speed is too slow, much more servers need to be rented to guarantee the computing capacity, and the increasing rent might surpass the savings of energy consumption cost, which also reduce the net profit.

7.1.3 Optimal Speed and Size

The third group of numerical calculations are to find the optimal server size m and server speed s under the given λ_{max}, ϱ, d, u, σ, α, β, γ, ξ, P^* and $\overline{r}$ such that the profit is maximized. Here, we use the same parameters in Fig. 7 and Fig. 8.

In Table 1, we demonstrate the optimal server size, optimal server speed, and maximal profit as a function of λ_{max}, respectively. The values are given for $7 \leq \lambda_{max} \leq 31$ in step of 2. From Table 1 it is noticed that the optimal server size is monotonically increasing with the increase of λ_{max}. That is because higher market demand requires greater computing capacity.

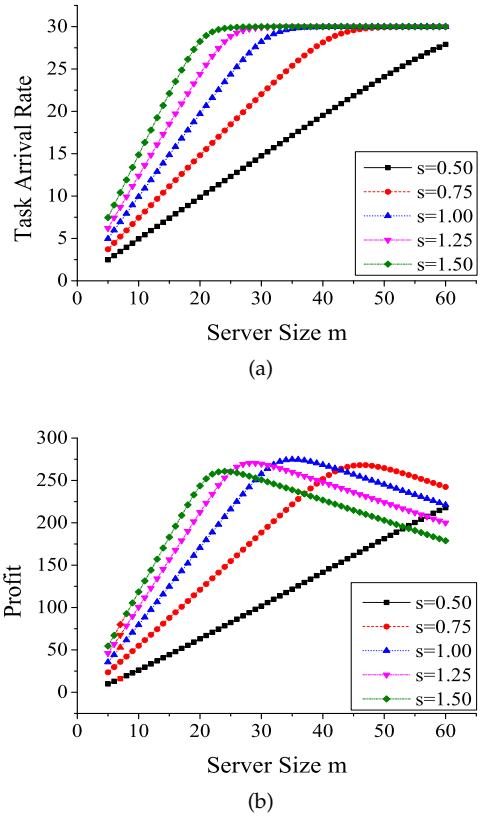


Fig. 9: Optimal profit and the number of invested domains versus total investment.

7.2 Actual Task Arrival Rate

In this paper, we consider the customer satisfaction in profit optimization configuration problem due to the affection of customer satisfaction on the task arrival rate. In order to verify how the task arrival rate changes with the varying configuration, we conduct the following numerical calculations. Here, the total market demand is set as $\lambda_{max} = 30$, and the other parameters are same with those in Fig. 7, 8.

In this group of numerical calculations, the task arrival rate is demonstrated as a function of m and s. Here, the total market demand is set as $\lambda_{max} = 30$, and the other parameters are same with those in Fig. 7 and Fig. 8. For $s = 0.5, 0.75, 1.0, 1.25, 1.5$, we display the actual task arrival rate and the corresponding profit for $5 \leq m \leq 30$, respectively. Fig. 9(a) shows that with the increase of server size and speed, the task arrival rate keeps an ascending trend as a whole. That is because increasing the number of servers or speeding up the servers can improve computing capacity. Hence customers can be served better, which leads to higher satisfaction. Due to the linear relationship between customer satisfaction and task arrival rate, the task arrival rate is increasing correspondingly. In addition, from Fig. 9(a) it is known that after the service capacity reaches a certain level, neither scaling up the server size nor speeding up the servers can increase the task arrival rate further. That is because the total market demand is limited and the actual task arrival rate cannot exceed it.

Fig. 9(b) gives the changing trend of profit, which can be explained properly by the changing trend of task arrival rates in Fig. 9(a). The figure shows that the profit firstly increases with the increasing number of servers. Yet when the server size reaches a certain point, increasing server size no longer produces more profit but a loss of profit. That is because the market demand is limited, increasing the number of servers further only generate unnecessary cost while won't increase the revenue. Hence, the profit decreases of course.

7.3 Performance Comparison

In this section, we compare the task arrival rate and the profit under the configuration determined by the model of [2] and ours. In the model proposed in [2], the task arrival rate is taken as a constant, and the optimal configuration is calculated based on the known task arrival rate. However, according to our analysis, the task arrival rate is affected by the customer satisfaction level, so the actual task arrival rate must be less than the known value, and its actual profit is less than the expected optimal profit. In our profit maximization model, we take into consideration the affection of customer satisfaction on task arrival rate. Hence, the optimal configuration is different from [2]. In Table 2, the optimal configurations of two models and corresponding profit are presented. In this group of calculations, the λ_{max} is changing from 5 to 35 in step of 5.

From the table, we can see that for all λ_{max} values, both of the actual task arrival rate and profit with satisfaction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2667706, IEEE Transactions on Sustainable Computing

TABLE 1: The optimal configuration and the corresponding profit

λmax	7	9	11	13	15	17	19	21	23	25	27	29	31
Optimal Size	9	11	13	15	17	19	21	23	25	27	29	31	33
Optimal Speed	1.02	1.03	1.04	1.05	1.05	1.05	1.05	1.06	1.06	1.06	1.06	1.06	1.06
Maximal Profit	60.7437	79.3505	98.0317	116.7723	135.5658	154.3989	173.2676	192.1713	211.1052	230.0634	249.0433	268.0426	287.0596

TABLE 2: Comparison of Optimal Solutions

λ_{max}	Without Considering Satisfaction				Considering Satisfaction			
	optM	optS	λ_{act}	Profit$_{act}$	optM	optS	λ_{act}	Profit$_{act}$
5	6.05	1.05	4.7813	42.0361	6	1.13	4.8532	42.3349
10	11.67	1.01	9.6332	87.6533	12	1.04	9.7933	88.6833
15	17.23	0.99	14.5521	134.7460	17	1.05	14.7198	135.5658
20	22.76	0.98	19.4371	181.4381	22	1.06	19.6815	182.7143
25	28.27	0.98	24.4340	229.4875	27	1.06	24.6299	230.0634
30	33.78	0.97	29.3031	276.3778	32	1.06	29.5876	277.5490
35	39.27	0.97	34.3053	324.6545	37	1.06	34.5527	325.1396

consideration are greater than that without satisfaction consideration. That is because the profit maximization model in [2] does not consider the effect of customer satisfaction on task arrival rate, and the optimal configuration is calculated based on λ_{max} but not the actual task arrival rate λ_{act}, so it is not the real optimal values. While in our profit maximization model, the optimal configuration is calculated based the actual task arrival rate, hence, it can generate more profit.

8 Conclusions

In this paper, we consider customer satisfaction in solving optimal configuration problem with profit maximization. Because the existing works do not give a proper definition and calculation formula for customer satisfaction, hence, we first give a definition of customer satisfaction leveraged from economics and develop a formula for measuring customer satisfaction in cloud. Based on the affection of customer satisfaction on workload, we analyze the interaction between the market demand and the customer satisfaction, and give the calculation of the actual task arrival rate under different configurations. In addition, we study an optimal configuration problem of profit maximization. The optimal solutions are solved by a discrete hill climbing algorithm. Lastly, a series of calculations are conducted to analyze the changing trend of profit. Moreover, a group of calculations are conducted to compare the profit and optimal configuration of two situations with and without considering the affection of customer satisfaction on customer demand. The results show that when considering customer satisfaction, our model performs better in overall.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments and suggestions. The research was partially funded by the Key Program of National Natural Science Foundation of China (Grant Nos. 61133005, 61432005), the National Natural Science Foundation of China (Grant Nos. 61370095, 61472124), National High-tech R&D Program of China (2015AA015303).

References

[1] P. Mell and T. Grance, "The nist definition of cloud computing," *Communications of the Acm*, vol. 53, no. 6, pp. 50–50, 2011.

[2] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, "Optimal multiserver configuration for profit maximization in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1087–1096, 2013.

[3] "Amazon EC2," http://aws.amazon.com, 2015.

[4] "Microsoft Azure," http://www.microsoft.com/windowsazure, 2015.

[5] "Saleforce.com," http://www.salesforce.com/au, 2014.

[6] J. Mei, K. Li, A. Ouyang, and K. Li, "A profit maximization scheme with guaranteed quality of service in cloud computing," *IEEE Trans. Computers*, vol. 64, no. 11, pp. 3064–3078, Nov 2015.

[7] R. N. Cardozo, "An experimental study of customer effort, expectation, and satisfaction," *Journal of marketing research*, pp. 244–249, 1965.

[8] J. A. Howard and J. N. Sheth, *The theory of buyer behavior*. Wiley New York, 1969, vol. 14.

[9] G. A. Churchill Jr and C. Surprenant, "An investigation into the determinants of customer satisfaction," *Journal of marketing research*, pp. 491–504, 1982.

[10] D. K. Tse and P. C. Wilton, "Models of consumer satisfaction formation: An extension," *Journal of marketing research*, pp. 204–212, 1988.

[11] A. Parasuraman, V. A. Zeithaml, and L. L. Berry, "Reassessment of expectations as a comparison standard in measuring service quality: implications for further research," *the Journal of Marketing*, pp. 111–124, 1994.

[12] K. Medigovich, D. Porock, L. Kristjanson, and M. Smith, "Predictors of family satisfaction with an australian palliative home care service: a test of discrepancy theory," *Journal of palliative care*, vol. 15, no. 4, p. 48156, 1999.

[13] J. J. Jiang, G. Klein, and C. Saunders, *Discrepancy Theory Models of Satisfaction in IS Research*. New York, NY: Springer New York, 2012, pp. 355–381.

[14] Y. Hu, J. Wong, G. Iszlai, and M. Litoiu, "Resource

provisioning for cloud computing," in *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*. IBM Corp., 2009, pp. 101–111.

[15] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers' revenues via energy aware allocation policies," in *2010 IEEE 3rd International Conference on Cloud Computing (CLOUD)*. IEEE, 2010, pp. 131–138.

[16] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[17] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Trans. Computers*, vol. 63, no. 1, pp. 45–58, 2014.

[18] S. Chaisiri, B.-S. Lee, and D. Niyato, "Profit maximization model for cloud provider based on windows azure platform," in *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, May 2012, pp. 1–4.

[19] S. Liu, S. Ren, G. Quan, M. Zhao, and S. Ren, "Profit aware load balancing for distributed cloud data centers," in *2013 IEEE 27th International Symposium on Parallel & Distributed Processing (IPDPS)*. IEEE, 2013, pp. 611–622.

[20] J. Chen, C. Wang, B. B. Zhou, L. Sun, Y. C. Lee, and A. Y. Zomaya, "Tradeoffs between profit and customer satisfaction for service provisioning in the cloud," in *Proceedings of the 20th international symposium on High performance distributed computing*. ACM, 2011, pp. 229–238.

[21] W. Gao and F. Kang, "Cloud simulation resource scheduling algorithm based on multi-dimension quality of service," *Information Technology Journal*, vol. 11, no. 1, pp. 94–101, 2012.

[22] L. Wu, S. K. Garg, and R. Buyya, "Sla-based admission control for a software-as-a-service provider in cloud computing environments," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1280–1299, 2012.

[23] K. C. Wu, H. C. Jiau, and K.-F. Ssu, "Improving consumer satisfaction through building an allocation cloud," in *The Fifth International Conference on Dependability (DEPEND 2012)*, 2012, pp. 31–37.

[24] C. Jing, Y. Zhu, and M. Li, "Customer satisfaction-aware scheduling for utility maximization on geo-distributed cloud data centers," in *2013 IEEE International Conference on HPCC_EUC*. IEEE, 2013, pp. 218–225.

[25] K. Tsakalozos, H. Kllapi, E. Sitaridi, M. Roussopoulos, D. Paparas, and A. Delis, "Flexible use of cloud resources through profit maximization and price discrimination," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, April 2011, pp. 75–86.

[26] R. Chen, Y. Zhang, and D. Zhang, "A cloud task scheduling algorithm based on users' satisfaction," in *2013 Fourth International Conference on Networking and Distributed Computing (ICNDC)*, Dec 2013, pp. 1–5.

[27] M. Unuvar, S. Tosi, Y. Doganata, M. Steinder, and A. Tantawi, "Selecting optimum cloud availability zones by learning user satisfaction levels," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[28] H. Morshedlou and M. Meybodi, "Decreasing impact of sla violations:a proactive resource allocation approach for cloud computing environments," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 156–167, April 2014.

[29] R. C. Lewis and B. H. Booms, "The marketing aspects of service quality," *Emerging perspectives on services marketing*, vol. 65, no. 4, pp. 99–107, 1983.

[30] U. Lehtinen and J. R. Lehtinen, *Service quality: a study of quality dimensions*. Service Management Institute, 1982.

[31] C. Grönroos, "A service quality model and its marketing implications," *European Journal of marketing*, vol. 18, no. 4, pp. 36–44, 1984.

[32] N. Kano, N. Seraku, F. Takahashi, and S. I. Tsuji, "Attractive quality and must-be quality," *Journal of the Japanese Society for Quality Control*, vol. 14, no. 2, pp. 39 – 48, April 1984.

[33] X. Chang, B. Wang, J. Muppala, and J. Liu, "Modeling active virtual machines on iaas clouds using an m/g/m/m+k queue," *IEEE Transactions on Services Computing*, pp. 408 – 420, 2014.

[34] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power cmos digital design," *IEICE Transactions on Electronics*, vol. 75, no. 4, pp. 371–382, 1992.

[35] S. Skiena and S. Skiena, *The Algorithm Design Manual (2. ed.)*. Springer, 2008.

[36] E. Sauerwein, F. Bailom, K. Matzler, and H. H. Hinterhuber, "The kano model: How to delight your customers," *IX International Working Seminar on Production Economics*, vol. 5, no. 1, pp. 6 – 18, 1996.

[37] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computingłthe business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.

[38] B. S. Skeina, "The algorithm design manual," no. 3, pp. 351–358, 2013.

[39] H. Goudarzi and M. Pedram, "Maximizing profit in cloud computing system via resource allocation," in *2011 31st International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2011, pp. 1–6.

[40] L. Kleinrock, *Theory, volume 1, Queueing systems*. Wiley-interscience, 1975.

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

[42] P. de Langen and B. Juurlink, "Leakage-aware multiprocessor scheduling," *Journal of Signal Processing Systems*, vol. 57, no. 1, pp. 73–88, 2009.

[43] J. Mei, K. Li, J. Hu, S. Yin, and E. H-M Sha, "Energy-aware preemptive scheduling algorithm for sporadic tasks on dvs platform," *Microprocessors and Microsystems*, vol. 37, no. 1, pp. 99–112, 2013.

[44] G. A. Churchill and C. Surprenant, "An investigation into the determinants of customer satisfaction." *Journal of Marketing Research*, vol. 19, no. 4, pp. 491–504, 1982.

[45] Q. Zheng and B. Veeravalli, "On the design of mutually aware optimalpricing and load balancing strategiesfor grid computing systems," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1802–1811, July 2014.

Jing Mei received the Ph.D in computer science from Hunan University, China, in 2015. She is currently an assistant professor in the College of Mathematics and Computer Science at Hunan Normal University. Her research interests include parallel and distributed computing, cloud computing, etc. She has published 9 research articles in international conference and journals, such as IEEE Transactions on Computers, IEEE Transactions on Service Computing, Cluster Computing, Journal of Grid Computing, Journal of Supercomputing.

Kenli Li received the PhD degree in computer science from Huazhong University of Science and Technology, China, in 2003. He was a visiting scholar at University of Illinois at Urbana Champaign from 2004 to 2005. He is currently a full professor of computer science and technology at Hunan University and deputy director of National Supercomputing Center in Changsha. His major research includes parallel computing, cloud computing, and DNA computing. He has published more than 100 papers in international conferences and journals, such as IEEE-TC, IEEE-TPDS, IEEE-TSP, JPDC, ICPP, CCGrid, FGCS. He is an outstanding member of CCF.

Keqin Li is a SUNY Distinguished Professor of computer science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published over 460 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Sustainable Computing*.. He is an IEEE Fellow.