User Perceived Value-Aware Cloud Pricing for Profit Maximization of Multiserver Systems

Peijin Cong[†], Liying Li[†], Gaoyuan Shao[†], Junlong Zhou[*], Mingsong Chen[†], Kai Huang[‡], and Tongquan Wei[†§]
[†]Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China
[*]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[‡]School of Data Engineering and Computer Science, Sun Yat-sen University, Guangzhou 510275, China

***Abstract*—With the rapid deployment of cloud computing infrastructures, understanding the economics of cloud computing has becoming a pressing issue for cloud service providers. However, existing pricing models rarely consider the dynamic interaction between user requests and the cloud service provider, thus can not accurately reflect the law of supply and demand in marketing. In this paper, we propose a pricing model based on the concept of user perceived value in the domain of economics that accurately capture the real supply and demand situation in the cloud service market. We then design a profit maximization scheme based on the presented dynamic pricing model that optimizes profit of the cloud service provider without violating user service-level agreement. Extensive experiments using data extracted from real-world applications validate the effectiveness of the proposed user perceived value-based pricing model. The proposed profit maximization scheme achieves 24.44% more profit as compared to the state of the art benchmarking methods.**

***Index Terms*—Cloud computing, dynamic pricing model, user perceived value, profit maximization.**

I. Introduction

Cloud computing has become an effective commercial computing model that distributes user requests on a pool of servers and delivers hosted services over Internet. As a business model, it turns resources of computing, storage, and communication into ordinary commodities and utilities in a pay-as-you-go manner [1]–[4]. It is natural for cloud service providers to pursue the goal of profit maximization, thus, the cloud service pricing strategy is of particular importance to cloud service providers.

The pricing model of a cloud service provider in cloud computing consists of two parts, that is, the revenue and the cost [5]. From the perspective of a cloud service provider, the revenue is the income that the cloud service provider has from the sale of cloud services to users, and the cost is the expenditure of renting and electricity bill of the server systems. To pursue profit maximization, cloud service providers attempt to increase revenue by setting a high price for cloud services and attracting a great amount of service purchase. However, service price and purchase amount interplay and cannot be optimized simultaneously [6]. On the other hand, the cost needs to be reduced for profit maximization, thus, aspects such as multiserver configurations and electricity price should be considered in pricing modeling.

This work was partially supported by Shanghai Municipal Natural Science Foundation (Grant No. 16ZR1409000) and Natural Science Foundation of China (Grant No. 61672230). [§]Corresponding author: tqwei@cs.ecnu.edu.cn.

Numerous investigations have been made into pricing mechanisms for profit maximization in cloud computing. Fixed pricing strategies such as pay-per-use, subscription based pricing, and tiered pricing are the most common pricing methods used by major cloud service providers [5], [7], [8]. However, these pricing methods cannot meet the dynamic needs of users and cannot reflect the market situation of supply and demand, which necessitates dynamic pricing strategies that adjust price of cloud services according to market situation and user requirements for service quality. Macias et al. [9] proposed a genetic model based pricing strategy that obtains optimal pricing in an iterative way. Amazon [10], [11] utilizes a spot pricing strategy that dynamically adjusts prices for a virtual service instance to accommodate changes in supply and demand. Cao et al. [5] presented a pricing model that takes such factors into considerations as the configuration of a multiserver system, the service-level agreement, the satisfaction of a consumer, and a cloud service provider's margin and profit. Though these works investigate dynamic pricing strategies from different perspectives, the interaction between users and cloud service providers with respect to supply and demand relationship is not discussed.

In this paper, we propose a user perceived value-based pricing mechanism that conforms to the law of supply and demand in economics. The novel contributions of this paper are summarized as follows:

- We propose a dynamic pricing model that considers the interplay between cloud users and cloud service providers. The model built upon the concept of user perceived value in the domain of economics accurately captures the dynamics of supply and demand in cloud pricing strategies.
- We propose a profit maximization scheme based on the presented dynamic pricing model. The proposed scheme optimizes the profit of cloud service providers by configuring multiservers systems under the constraint of user service-level agreement.
- Extensive simulation experiments show that the proposed scheme is superior to two benchmarking pricing models. The proposed scheme can obtain up to 10.748 cents per second more as compared to benchmarking methods.

The remainder of the paper is organized as follows. Section II presents the system architecture and models, Section III describes the proposed user perceived value-based pricing mechanism. The effectiveness of the proposed scheme is

validated in Section IV and concluding remarks are given in Section V.

II. System Architecture and Models

We consider a common three-tier cloud service provision structure that consists of cloud users, cloud service providers, and cloud infrastructure vendors [5], [8], [12]. Among the three entities that form a market in the cloud computing, the infrastructure vendor charges the cloud service provider for renting infrastructures to deploy service capacity, and the cloud service provider charges cloud users for processing their requests. In this paper, cloud user and cloud service provider are of our particular interest. We will introduce our cloud user model and cloud service provider model in the following sections.

A. Cloud User Model

To maximize the profit of a cloud service provider, the cloud service provider needs to know the aggregate demands of all users in the market. When a cloud service provider sets up the price of a service, different users have different responses to this price. As with conventional market commodities, the cloud computing service can be seen as a special commodity which follows market rules. That is, the price of cloud computing service is also dictated by the supply and demand in the market. In this paper, we propose a perceived value oriented pricing strategy for cloud computing services. In this subsection, we introduce the concepts of user perceived value and user request (or demand) distribution.

User Perceived Value: In conventional markets, the arrival rate of customers to a store is often a response to their regular purchasing patterns rather than a reaction to individual prices [6]. Thus, it is reasonable to assume that the change of the list price has no effects on the total number of customers who are visiting the store. Typically, not all of the customers have the willing to buy a specific commodity, that is, the total number of customers who buy commodities are no larger than the total number of people that visit the store.

Customer perceived value is defined as the worth that a product or service has in the mind of a consumer. In general, customers are unaware of the true cost of production for the products they buy, instead, they simply have an internal feeling for how much certain products are worth to them. In the conventional market environment, only the customer whose perceived value is higher than the real price of the product is willing to pay for the product.

In this paper, we adopt the terminology of customer perceived value used in traditional market environment. We take the cloud computing environment as a store and the cloud computing service is deemed as a special commodity provided in the store. In the following sections, the terminology of customer perceived value and user perceived value are used interchangeably.

User Demand Distribution: Unlike traditional methods that use the expected demand to model user behavior [13], [14], we use the probability distribution of the total demands in this work to model user requests.

We consider a slotted time model that deals with the pricing decision and constraints for discrete time intervals (also called sales periods) of equal length τ. Specifically, a cloud service provider sets list price for the service at the beginning of regular sales periods. The list price during each sales period is assumed to be constant, but varies from period to period.

Suppose that the cloud service provider will charge ω per user for a specific cloud service S during a sales period τ. Let n denote the total number of users that have interest in the service at the price of ω during the sales period τ, and λ_u denote the number of users arriving per unit time, respectively. The n is assumed to be independent of all other parameters of the system, and is a discrete Poisson random variable distributed as

$$P(n|\lambda_u) = \frac{(\lambda_u \tau)^n e^{-\lambda_u \tau}}{n!},\ n = 0, 1, 2, \cdots, \infty. \quad (1)$$

However, the user arrival rate λ_u may not be constant in many situations. Taking into account the heterogeneity of arrival rate, a Gamma distribution characterized by parameters (α, β) is utilized to represent the arrival rate λ_u, the probability density function of which is given by

$$g(\lambda_u) = \frac{1}{\Gamma(\alpha)\beta^\alpha} \lambda_u^{(\alpha-1)} e^{-\lambda_u/\beta},\ 0 \le \lambda_u \le \infty, \quad (2)$$

where the expectation and variance of λ_u is given by $E[\lambda_u] = \alpha\beta$ and $Var[\lambda_u] = \alpha\beta^2$, respectively, and $\Gamma(\alpha)$ is a complete gamma function.

Among the n users, any one whose perceived value of the service is no less than the list price ω is considered as a potential buyer of the service. Let m denote the number of potential buyers. It is a non-negative discrete random variable taking the value of $0, 1, 2, \cdots, \infty$ and $m \le n$ holds. Let X_i denote the perceived value that user i has for the service S. X_i is a continuous random variable and $0 \le X_i < \infty$ holds. As with other benchmarking pricing models [14], $X_1, X_2, \cdots, X_n$ are assumed to be independent and identical random variables. The probability density function of the random variables, denoted by $f(x)$, is known or can be estimated a priori. Let $F(\omega)$ represent the cumulative distribution function of x evaluated at ω. The $F(\omega)$ is a non-decreasing function of ω, and $0 \leqslant F(\omega) \leqslant 1$ and $\lim_{\omega \to \infty} F(\omega) = 1$ hold [15]. Let $P_\omega(m|n)$ indicate the probability that m out of n users are inclined to buy in the sales period when the service price is set equal to ω. It follows a binomial distribution of probability, which is given by

$$P_\omega(m|n) = \binom{n}{m} [1 - F(\omega)]^m [F(\omega)]^{(n-m)}. \quad (3)$$

Combining (1)-(3), we can derive the probability of having m potential buyers during the sales period τ when the service price is set equal to ω. The probability is denoted by $P_\omega(m)$ and given by

$$\begin{aligned} P_\omega(m) &= \int_{\lambda_u=0}^{\infty} \sum_{n=0}^{\infty} P_\omega(m|n) P(n|\lambda_u) g(\lambda_u) d\lambda_u \\ &= \binom{m+\alpha-1}{m} \left[\frac{\beta\tau[1 - F(\omega)]}{1 + \beta\tau[1 - F(\omega)]}\right]^m \left[\frac{1}{1 + \beta\tau[1 - F(\omega)]}\right]^\alpha. \end{aligned} \quad (4)$$

Clearly, it is a negative binomial distribution. As a result, the expected number of actual buyers of the service at price ω during sales period τ, which is denoted by $E_\omega(m)$, can be calculated as

$$E_\omega(m) = \alpha\beta\tau(1 - F(\omega)), \tag{5}$$

where α and β are parameters of the Gamma distribution of user arrival rate λ_u, and $F(\omega)$ is the cumulative distribution function of x evaluated at ω. The revenue of the cloud service provider in a sales period τ is thus given by

$$Revenue = \omega \times E_\omega(m) = \omega\alpha\beta\tau(1 - F(\omega)). \tag{6}$$

B. Cloud Service Provider Model

The cloud service provider rents a multiserver system that is constructed and maintained by an infrastructure vendor to serve user requests. The architecture details of the multiserver system are quite flexible [5], [16]. They can be blade centers where each server is a server blade [17], clusters of traditional servers where each server is an ordinary processor [18], and multicore server processors where each server is a single core [19]. For the sake of easy presentation, these blades/processors/cores are simply called servers. Cloud users of a cloud service provider submit their requests to the cloud service provider, and the cloud service provider serves these requests (i.e., run these tasks) on the multiserver system.

Multi-Server Model: We consider a multiserver system that consists of M homogeneous servers operating at a common speed of s. The multiserver system can be modeled as an M/M/M queuing system where arrivals of user requests governed by a Poisson process form a single queue and M servers can process these requests in parallel. Let μ be the service rate of user requests that arrive at the rate of λ_u. It is clear that μ user requests can be processed by servers if the number of user requests in the system is not greater than M. The service time of a user request on a server is an exponential random variable denoted by $x_1 = r/s$ with mean $\overline{x_1} = \overline{r}/s$, where r is the number of instructions to be executed for the service request. A first-come-first-served (FCFS) queue of infinite capacity is maintained by the multiserver system for waiting tasks when all the servers are busy. Let ρ be server utilization, which is defined as the average percentage of time that a server is busy. It can be expressed as

$$\rho = \frac{\lambda_u}{M\mu} = \frac{\lambda_u}{M\frac{s}{\overline{r}}} = \frac{\lambda_u\overline{r}}{Ms}. \tag{7}$$

Let P_k be the probability of k service requests being waiting or processing in the M/M/M queuing system. Based on queuing theory [5], [20], P_k is given by

$$P_k = \begin{cases} P_0\frac{(M\rho)^k}{k!}, & k \le M \\ P_0\frac{M^M\rho^k}{M!}, & k \ge M \end{cases}, \tag{8}$$

where P_0 is the probability that there are no tasks in the queue, and is formulated into [20]

$$P_0 = (\sum_{k=0}^{M-1}\frac{(M\rho)^k}{k!} + \frac{(M\rho)^M}{M!}\cdot\frac{1}{1-\rho})^{-1}.$$

The probability that there are exact M service requests in the system is thus given by $P_M = P_0\frac{(M\rho)^M}{M!}$. By using the Taylor series expansions of $\sum_{k=0}^{M-1}(M\rho)^k/k! \approx e^{M\rho}$ and $M! \approx \sqrt{2\pi M}(\frac{M}{e})^M$, it could be transformed into

$$P_M = \frac{1-\rho}{\sqrt{2\pi M}(1-\rho)(\frac{e^{\rho-1}}{\rho})^M + 1}. \tag{9}$$

This form of P_M is necessary to derive multiserver configurations in Section III.

When all the servers in the system are busy, a newly submitted service request must wait and will be inserted into the FCFS queue. Let P_q denote the probability of queuing a newly arrived task when no servers are idle at the time of arrival. P_q can be formulated as

$$P_q = \sum_{k=M}^{\infty} P_k = \frac{P_M}{1-\rho}. \tag{10}$$

Let $\overline{N}$ denote the average number of service requests being waiting or executing in the multiserver system, then $\overline{N}$ is calculated as

$$\overline{N} = \sum_{k=0}^{\infty} kP_k = M\rho + \frac{\rho}{1-\rho}P_q. \tag{11}$$

The average service response time $\overline{R}$ which is defined as the time elapsed between the time when a request is submitted to the time when the request is finished, is adopted to evaluate the service quality. It is in fact the sum of task execution time and waiting time, and can be derived by applying Little's Law [21] as

$$\overline{R} = \frac{\overline{N}}{\lambda_u} = \overline{x_1}(1 + \frac{P_q}{M(1-\rho)}) = \overline{x_1}(1 + \frac{P_M}{M(1-\rho)^2}). \tag{12}$$

The average service response time $\overline{R}$ is utilized in this paper as a metric for service-level agreement. If the response time of a service exceeds the predefined deadline, the service-level agreement is deemed to be violated.

Gross Profit: The gross profit a cloud service provider earns is the total revenue subtracted by the cost of generating that revenue. In other words, gross profit is sales minus cost of the cloud service sold. Assuming the price of cloud service is constant in a sales period, the revenue earned is given by $\omega \cdot E_\omega(m)$, where ω denotes the service price per user and $E_\omega(m)$ indicates the expected number of actual buyers at price ω during the sales period.

The cost of cloud service sold mainly consists of the cost paid to rent cloud computing infrastructure, and the electricity expense incurred by the cloud service provider to maintain the operation of the computing infrastructure. Let δ be the fee a cloud service provider pays to rent a server during a sales period, the rent the cloud service provider needs to pay for a system of M servers during the sales period is

$$Rent = M\delta. \tag{13}$$

As a portion of the cloud service cost, electricity fee has become a significant expense for today's data centers. It can

be derived by multiplying energy consumed by a server with electricity price. The energy consumed by a server can be modeled at different levels of abstraction. At the abstraction level of digital CMOS circuit, the power consumption, which is denoted by P_{tot}, can be modeled as

$$P_{tot} = P_{sta} + P_{dyn}, \tag{14}$$

where P_{sta} is the static power dissipation while P_{dyn} is the dynamic power dissipation. P_{sta} is independent of switching activity and maintains the basic circuit state, thus can be deemed as a constant [5]. P_{dyn} is related to processor switching activity and dominates the total power consumption, which can be formulated as a function of supply voltage v and processing speed s. In addition, the supply voltage is usually linearly proportional to the processing speed, that is, $v \propto s$. The dynamic power consumption P_{dyn} is then expressed as ξs^{γ}, where ξ is a processor dependent coefficient and γ is a constant that equals to $2\phi + 1$ ($\phi > 0$). Based on the static and dynamic power consumption described above, we use the following Equation (15) to denote the total power consumption of a multiserver system, that is,

$$P_{tot} = M((P_{dyn} - P_{sta})\rho + P_{sta}), \tag{15}$$

where M is the number of servers and ρ is the server utilization.

Let E^{τ} denote the energy consumed by all M servers in the system during the sales period τ. Then it is given by

$$E^{\tau} = M((P_{dyn} - P_{sta})\rho + P_{sta}) \times \tau. \tag{16}$$

Let $C^{\tau}(E^{\tau})$ denote the price of the energy consumed by all servers in the sales period τ, then $C^{\tau}(E^{\tau})$ can be formulated as

$$C^{\tau}(E^{\tau}) = \begin{cases} k_1^{\tau}, & 0 \le E^{\tau} \le l_{th}^{\tau} \\ k_2^{\tau}, & E^{\tau} > l_{th}^{\tau} \end{cases} \tag{17}$$

where $k_1^{\tau}, k_2^{\tau} > 0$ are differentiated price and l_{th}^{τ} is the energy consumption threshold in the sales period τ. The electricity bill of the multiserver system in the sales period τ is hence formulated as

$$\begin{aligned} Bill &= E^{\tau} \times C^{\tau}(E^{\tau}) \\ &= M((P_{dyn} - P_{sta})\rho + P_{sta}) \times \tau \times C^{\tau}(E^{\tau}). \end{aligned} \tag{18}$$

We define the profit of the cloud service provider in a sales period τ as the revenue minus the various expenses including the electricity cost and rental cost incurred in the sales period, that is,

$$Profit = Revenue - Bill - Rent, \tag{19}$$

where $Revenue$, $Bill$, and $Rent$ are given in Equations (6), (18), and (13), respectively.

III. User Perceived Value-Aware Profit Optimization Scheme

A. Problem Definition

The price of a cloud service interplays with the number of users who purchase the service, which in turn affects the revenue of the cloud service provider. This paper aims to maximize the profit of the cloud service provider by deriving the optimal number of servers, operating speed of servers, and price of services provided without violating the user service-level agreement. In addition to the user service-level agreement, the power consumed by the multiserver system can not exceed a threshold value.

We assume that the cloud service provider optimizes its decisions at the beginning of each sales period τ. Let b_1 denote the upper bound on the power consumption of the M servers, and b_2 be the upper bound on the expected response time of user requests. The optimization problem we will solve is thus formulated into

$$\begin{aligned} \text{Maximize: } & Profit \\ \text{subject to: } & P_{tot} \le b_1 \\ & \overline{R} \le b_2 \end{aligned} \tag{20}$$

where $Profit$ of the cloud service provider, power consumption P_{tot} of the multiserver system, and service-level agreement metric $\overline{R}$ are given in Equations (19), (15), and (12), respectively. The optimization problem tries to maximize the $Profit$ of the cloud service provider under constraints of power budget of the multiserver system and expected delay of user requests.

B. Create Augmented Lagrangian Function

The problem given in (20) is convex since the objective function $Profit$ and the constraints P_{tot} and $\overline{R}$ are all convex. Numerous techniques on constrained optimization have been investigated in the literature [22]–[24]. Of these techniques, the method of augmented Lagrange multipliers is a powerful tool for solving this class of problems, thus, is adopted in this work to solve the profit maximization problem given in Equation (20).

The $Bill$ given in Equation (18) is a function of power consumption of the multiserver system, the length of the sales period τ, and real-time price of electricity. Since real-time price is flat within each sales period τ and τ itself is constant, the $Bill$ for τ is fixed and can be expressed as $Bill = b_3 P_{tot}$, where P_{tot} given in Equation (15) is the total power consumed by the multiserver system and b_3 is a constant coefficient. The optimization problem given in Equation (20) can then be rewritten as

$$\begin{cases} O(\omega, M, s) = \omega E_{\omega}[m] - b_3 P_{tot} - \delta M \\ g_1(M, s) = b_2 - \overline{x_1}(1 + \frac{P_M}{M(1-\rho)^2}) \ge 0 \\ g_2(M, s) = b_1 - M((\xi s^{\gamma} - P_{sta})\rho + P_{sta}) \ge 0 \end{cases} \tag{21}$$

where $O(\omega, M, s)$ denotes the objective function of $Profit$ given in Equation (19), and $g_1(M, s)$ and $g_2(M, s)$ are constraint equations of M and s, respectively.

Next, we transform the optimization problem given in Equation (21) with inequality constraints into an augmented Lagrangian function. Let $\mathbf{y}$ be the vector that converts the optimization problem with inequality constraints to an optimization problem with equality constraints, and $\mathbf{v}$ be the Lagrange multiplier vector, the augmented Lagrangian function is thus

given by

$$\phi(\omega, M, s, \mathbf{y}, \mathbf{v}, \sigma) = O(\omega, M, s) - \sum_{j=1}^{2} v_j(g_j(M, s) - y_j^2) + \frac{\sigma}{2}\sum_{j=1}^{2}(g_j(M, s) - y_j^2)^2, \quad (22)$$

where the constant parameter σ denotes the penalty factor and $\sigma > 0$ holds. The augmented Lagrangian function given in Equation (22) can be converted into the form of

$$\phi(\omega, M, s, \mathbf{y}, \mathbf{v}, \sigma) = O(\omega, M, s) + \sum_{j=1}^{2}[\frac{\sigma}{2}[y_j^2 - \frac{1}{\sigma}(\sigma g_j(M, s) - v_j)]^2 - \frac{v_j^2}{2\sigma}] \quad (23)$$

by using the method of completing the square, a technique to derive the quadratic formula [25], and the function given in (23) can be easily maximized when

$$y_j^2 = \frac{1}{\sigma}\max(0, \sigma g_j(M, s) - v_j), j = 1, 2. \quad (24)$$

Plugging y_j^2 given in Equation (24) back into the original Formula (22), we have the desired augmented Lagrangian function

$$\phi(\omega, M, s, \mathbf{v}, \sigma) = O(\omega, M, s) + \frac{1}{2\sigma}\sum_{j=1}^{2}[[\max(0, v_j - \sigma g_j(M, s))]^2 - v_j^2]. \quad (25)$$

In other words, we convert the optimization problem given in Equation (21) with inequality constraints into the optimization problem given in Equation (25) without constraints. We seek to solve the augmented Lagrangian function given in Equation (25) by first computing the partial derivatives of ω, M, and s. The process is omitted here due to space limitation.

C. Solve Augmented Lagrangian Function

We present in this section an augmented Lagrangian method based algorithm that iteratively solves the optimization problem given in (22) and derives the optimum solution to service pricing and multiserver configurations. The proposed algorithm first computes an optimum Lagrangian multiplier, after which the optimal service pricing and multiserver configurations are determined.

Let $M^{(k)}$, $s^{(k)}$, and $v^{(k)}$ indicate the k^{th} iteration of M, s, and v in the algorithm. Let ε, η, and Ψ be three positive numbers, l be the number of iterations, and L be the maximum number of iterations. Algorithm 1 describes the proposed augmented Lagrangian algorithm. Inputs to the algorithm are electricity price C^τ during sales period τ, the rent δ, and user requests arrival rate λ_u. The algorithm iteratively derives the optimal service pricing ω and multiserver configurations including the optimal number of servers M, the server speed s, and the $Profit$ of the cloud service provider.

Algorithm 1: Iteratively solve the augmented Lagrangian function

Input:
Electricity price C^τ during sales period τ, rent δ, user requests arrival rate λ_u;

Output:
The optimal service price ω, number of servers M, server speed s, and $Profit$;

1 Formulate optimization problem into the form in Equation (25);
2 Set parameters α, β, γ, ε, η, Ψ, and L;
3 Initialize $M^{(0)}$, $s^{(0)}$, $v^{(1)}$, and l = 1;
4 **while** $l < L$ **do**
5 $[\omega^{(l)}, M^{(l)}, s^{(l)}] =$ **ALF-Solver**$(\phi(\omega, M^{(l-1)}, s^{(l-1)}, v^{(l)}, \sigma))$;
// Exit when $\{v^{(l)}\}$ converges;
6 **if** $\|Q(M^{(l)}, s^{(l)})\| < \varepsilon$ **then**
7 **break**;
8 **end**
// Otherwise, increase penalty factor σ;
9 **else if** $\|Q(M^{(l)}, s^{(l)})\|/\|Q(M^{(l-1)}, s^{(l-1)})\| \geq \Psi$ **then**
10 $\sigma = \eta\sigma$;
11 **end**
// Update the multiplier vector **v**;
12 $v_j^{l+1} = \max(0, v_j^l - \sigma g_j(M^{(l)}, s^{(l)}))(j = 1, 2)$;
13 $l = l + 1$;
14 **end**
15 Calculate the $Profit$ using the Equation (19);
16 **return** $[\omega^{(l)}, M^{(l)}, s^{(l)}, Profit]$;
17 **ALF-Solver**$(\phi(\omega, M^{(l-1)}, s^{(l-1)}, v^{(l)}, \sigma))$
18 Compute partial derivatives of ϕ w.r.t. ω, M, and s as $\partial\phi(\omega, M^{(l-1)}, s^{(l-1)}, v^{(l)}, \sigma)/\partial(\omega, M, s)$;
19 Calculate ω, M, and s based on a system of equations of $\frac{\partial\phi}{\partial\omega}$, $\frac{\partial\phi}{\partial M}$, and $\frac{\partial\phi}{\partial s}$;
20 **return** $[\omega, M, s]$;

The algorithm works as follows. It first formulates the optimization problem in the form as given in Equation (25), then sets parameters of ε, η, Ψ, and L, and initializes variables of $M^{(0)}$, $s^{(0)}$, v^1, and l (lines 1-3). In each round of iteration, the algorithm calls the augmented Lagrangian function solver, denoted by **ALF-Solver**$(\phi(\omega, M^{(l-1)}, s^{(l-1)}, v^{(l)}, \sigma))$, to obtain a local optimum of the ω, M, and s (line 5). The **ALF-Solver**$(\phi(\omega, M^{(l-1)}, s^{(l-1)}, v^{(l)}, \sigma))$ derives the local optimum by computing partial derivatives of $\phi(\omega, M, s, \mathbf{v}, \sigma)$ with regard to ω, M and s, and solving a system of equations of ω, M, and s (lines 17-20).

The algorithm exits if the Lagrangian multiplier vector $\mathbf{v}$ converges and approximates the optimum by an error of ε. Let $Q_j(M^{(l)}, s^{(l)}) = g_j(M^{(l)}, s^{(l)}) - y_j^2$ for $j = 1, 2$ be the penalty item of the augmented Lagrangian function given in Equation (22), then the Lagrangian multiplier vector $\mathbf{v}$ converges if $\|Q(M^{(l)}, s^{(l)})\| < \varepsilon$ holds (lines 6-8). If it does not converge or converges too slowly, that is, $\|Q(M^{(l)}, s^{(l)})\|/\|Q(M^{(l-1)}, s^{(l-1)})\| \geq \Psi$ holds for a positive number Ψ, the penalty factor σ is updated to $\eta\sigma$ for $\eta > 1$ to speed up the convergence process (lines 9-11). Accordingly, the Lagrangian multiplier for the next iteration is updated to $v_j^{l+1} = \max(0, v_j^{(l)} - \sigma g_j(M^{(l)}, s^{(l)}))(j = 1, 2)$ (lines 12-13), and the procedure moves to the next iteration.

Once the algorithm converges, the optimum of the ω, M, and s are derived, and the optimum of $Profit$ of the cloud service provider can be calculated by using Equation (19) (line 15). Line 16 returns the optimal service pricing, multiserver configurations, and $Profit$ of the cloud service provider.

IV. Numerical Results

Extensive simulation experiments have been conducted to validate the effectiveness of the proposed scheme. We first describe simulation settings in details, then verify the effectiveness of the proposed user perceived value-based pricing model, followed by the validation of the optimal pricing and multiserver configurations and a comparison study with benchmarking schemes in terms of the profit of the cloud service provider.

A. Simulation Settings

The simulation experiments are conducted on a machine equipped with 2.56GHz Intel i7 quad-core processor and 8GB DDR4 memory, and running a Windows version of Matlab_x64. For the sake of a fair comparison, three types of user requests used in [26] are also adopted in the experiment. The requests of type 1 are delay-sensitive while the requests of type 2 and 3 are elastic. The data of type 1 were extracted from Youtube U.S. traffic from January 1, 2014 to January 31, 2014 [27]. The data of type 2 and 3 were extracted from GMaps and GMail U.S. traffic from January 1, 2014 to January 31, 2014 [27]. The one day ahead real-time pricing data released by Ameren Illinois Power Corporation at January 2014 are taken as the price input in the experiment [28]. We also assume a normally distributed user perceived value X with mean of 0 and variance of 0.22, that is, $X \sim N(0, 0.22)$ [6], [29].

B. Verify Perceived Value-Based Pricing Model

This subsection verifies the proposed perceived value-based service pricing model from the perspective of the law of supply and demand.

Profit Vs. Service Requirement: We first analyze the relationship between the service requirement in terms of the number of instructions, which is denoted by r, and the profit of the cloud service provider. In addition to parameters given in Subsection IV-A, we set the average service requirement denoted by $\overline{r}$ to 1 billion instructions. The number of severs M is initialized to 7, the speed of servers s is initialized to 1 billion instructions per second, and the static power consumption P_{sta} is set to 2W. The parameters of dynamic power consumption are assumed to be $\gamma = 2.0$ and $\xi = 9.4192$, and parameters of Gamma distribution are assumed to be $\alpha = 2.0$ and $\beta = 1.5$ [5].

Fig. 1(a) shows the relationship between the profit of the cloud service provider and user's service requirements $(0 \leq r \leq 3)$ in billion instructions when service requirement arrival rate λ_u is 16.15, 16.35, 16.55, 16.75, and 16.95 billions instructions per second, respectively.

As shown in Fig. 1(a), profit decreases as λ_u increases. This is because with the increase of λ_u, servers can not process user requests in time, leading to a higher response time and lower quality of service. As a result, user perceived value of the service decreases, and the profit decreases as well. It also can be seen from Fig. 1(a) that the profit increases as service requirements increase. This indicates that the cloud service usage is proportional to the profit obtained under the perceived value-based pricing model.

Purchase Amount and Profit Vs. Service Price: Fig. 1(b) and Fig. 1(c) demonstrate that how the relationship among the cloud service purchase amount, profit, and the price of cloud service changes when λ_u is 16.75 and 16.95 billion instructions per second, respectively.

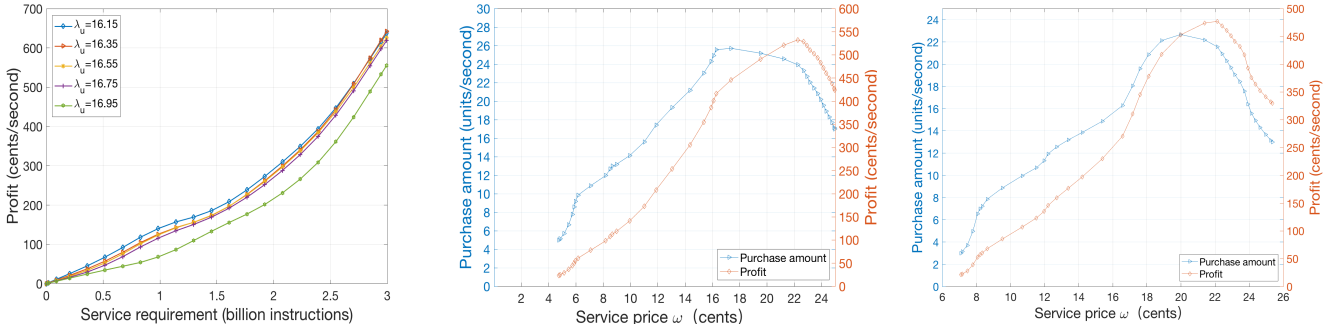
As we can see from Fig. 1(b) and Fig. 1(c), before the cloud service price reaches the perceived value of the service, the purchase amount of the cloud service increases with the increases of the price. Once the price exceeds the perceived value of the service, the purchase amount declines sharply. This observation is consistent with real market situation, that is, users are willing to accept a price and purchase when the price is lower than their perceived value. However, the user purchase intention will decline sharply when the price is beyond the user perceived value. It also can be seen from Fig. 1(b) and Fig. 1(c) that the point where purchase amount is maximum is not necessarily the point where the profit is maximum. That is, the profit for the scenario of the low price and high purchase amount is not necessarily higher than the profit for the scenario of the high price and low purchase amount.

C. Validate Multiserver Configurations for Profit Maximization

We set the response time constraint for user requests, denoted by b_1, to 0.33 seconds and the power consumption of the multiserver system, denoted by b_2, to 10^6W. The rental cost denoted by δ is set to 1.5 cents per second [6].

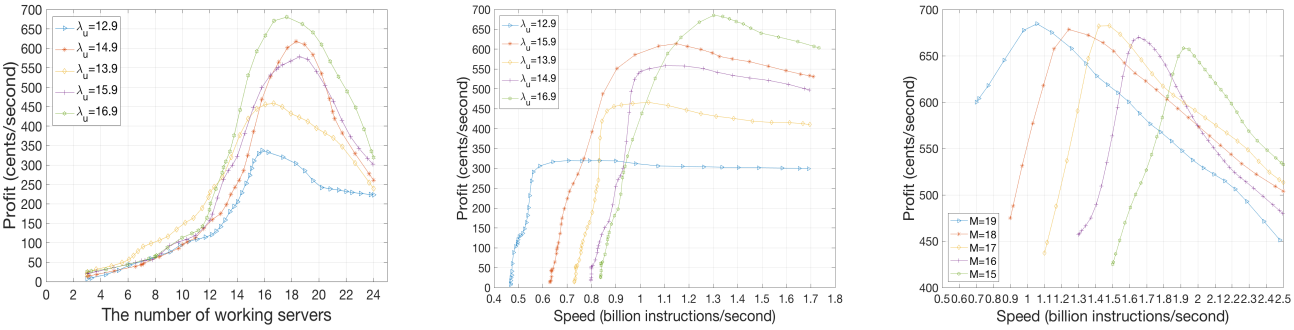
Fig. 2(a) shows the relationship between profit and the number of working servers. It can be seen from the figure that when user request arrival rate λ_u = 12.9, 13.9, 14.9, 15.9, and 16.9 billion instructions per second, the optimal number of servers denoted by M is 16, 17, 19, 18, and 17 respectively. It is clear that when M is small, the utilization of working servers is approaching 1, leading to a long response time for user requests, and in turn a low profit under the perceived value-based pricing model. As M increases, the number of user requests in the waiting queue decreases quickly, the user requests do not have to wait too long, and thus the profit increases under the perceived value-based pricing model. However, as M continues increasing, the profit does not increase. This is because the increase in the number of servers leads to an increase in the maintenance cost of working servers including electricity and rental cost.

Fig.2(b) shows the relationship between profit and the optimal server speed s. We notice from the figure that in order to maximize the profit, the optimal speed s is set to 0.7642, 0.9435, 1.1044, 1.1293, and 1.2838 billion instructions per second when the service request arrival rate λ_u = 12.9, 13.9, 14.9, 15.9, and 16.9 billion instructions per second,



(a) Service requirements vs. profit. (b) $\lambda_u = 16.75$. (c) $\lambda_u = 16.95$.

Figure 1: Verify perceived value-based pricing model.



(a) Profit vs. number of severs (M). (b) Profit vs. server speed (s). (c) Optimal server configuration (M, s).

Figure 2: Validate server configurations for profit maximization.

respectively. It is clear that when the server speed s is low, the utilization of working servers is approaching 1, leading to a long response time for user requests, and in turn a low profit under the perceived value-based pricing model. When the server speed s is high, service requirements are more likely to be met on time, leading to an increase in the profit under the perceived value-based pricing model. However, with the continued increase in s, the profit does not increase as expected. This is because increasing the server speed leads to an increase in the cost of operating a multiserver system.

From Fig. 2(a) and Fig. 2(b), we notice that profit reaches its maximum when λ_u is 16.9 billion instructions per second and the number and speed of servers take the appropriate value. Fig. 2(c) gives the optimal M and s of working servers that maximize the profit when λ_u = 16.9 billion instructions per unit time. It can be seen from Fig. 2(c) that the maximal profit is obtained when s and M is set to 1.4351 billion instructions per second and 17, respectively. That is to say, 687.9 cents of profit is obtained when 17 servers are open and each server runs at 1.4351 billion instructions per second.

D. Compare with Benchmarking Pricing Strategies

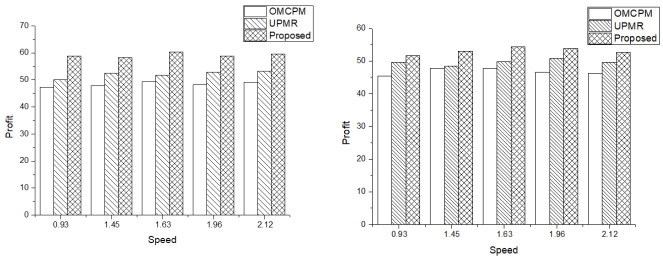
We compare the proposed perceived value-based profit maximization scheme with two benchmarking methods OMCPM [5] and UPMR [26]. OMCPM [5] is an efficient pricing model that takes such factors into considerations as the service-level agreement and customer satisfaction. It derives an optimal server configuration and service price for profit maiximzaiton. UPMR [26] is a usage-based pricing model used by today's major cloud operators. The UPMR model rewards users proportionally based on the time length that users set as deadlines for completing their workloads.

Two comparison experiments are conducted. In the first experiment, user request arrival rate λ_u is set to 16.9 billion instructions per second and the number of working servers M is set to 17. In the second experiment, λ_u is set to 12.55 billion instructions per second and M is set to 18.

We compare the maximal profit generated by proposed pricing model with that generated by the two benchmarking pricing model under the same experimental settings. It is clear from Fig. 3 that the proposed pricing model is superior to the two benchmarking models. For instance, the proposed pricing model can obtain up to 11.55 cents per second more (24.44%) as compared to OMCPM method, and 8.66 cents per second more (17.27%) as compared to UPMR when λ_u = 16.9 billion instructions per second, M = 17 and s = 0.93 billion instructions per second.

V. Conclusions

In this paper, we propose a user perceived value-based dynamic pricing model that takes into account the interplay between cloud users and cloud service providers. The profit maximization problem based on the proposed model is formulated into an augmented Lagrangian function and



(a) λ_u=16.9, M=17. (b) λ_u=12.55, M=18.

Figure 3: Compare with two benchmarking pricing models.

is iteratively solved using convex optimization techniques. Extensive experiments have been conducted to validate the effectiveness of the proposed scheme. The proposed profit maximization scheme can obtain more profit of up to 24.44% and 17.27% as compared to the state of the art benchmarking methods OMCPM [5] and UPMR [26], respectively.

REFERENCES

[1] K. Hwang, J. Dongarra, and G. Fox, Distributed and cloud computing, *Morgan Kaufmann*, 2012.

[2] L. Wang, D. Chen, Y. Hu, Y. Ma, and J. Wang, Towards enabling cyberinfrastructure as a service in clouds, *Computers and Electrical Engineering*, vol. 39, pp. 3-14, 2013.

[3] A. Khoshkbarforoushha, M. Wang, R. Ranjan, L. Wang, L. Alem, S. Khan, and B. Benatallah, Dimensions for evaluating cloud resource orchestration frameworks, *Computer*, vol. 49, pp. 24-33, 2016.

[4] J. Zhou, J. Chen, K. Cao, T. Wei, and M. Chen. Game theoretic energy allocation for renewable powered in-situ server systems, *IEEE International Conference on Parallel and Distributed Systems*, pp. 721-728, 2016.

[5] J. Cao, K. Hwang, K. Li, and A. Zomaya, Optimal multiserver configuration for profit maximization in cloud computing, *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1087-1096, 2013.

[6] Y. Chun, Optimal pricing and ordering policies for perishable commodities, *European Journal of Operational Research*, pp. 68-82, 2003.

[7] M. Ghamkhari and H. Mohsenian-Rad, Energy and performance management of green data centers: A profit maximization approach, *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1017-1025, 2013.

[8] Y. Lee, C. Wang, A. Zomaya, and B. Zhou, Profit-driven service request scheduling in clouds, *International Conference on Cluster, Cloud and Grid Computing*, pp. 15-24, 2010.

[9] M. Macias and J. Guitart, A genetic model for pricing in cloud computing markets, *ACM Symposium on Applied Computing*, pp. 113-118, 2011.

[10] Amazon EC2. [Online]. Available: http://aws.amazon.com.

[11] Amazon EC2 spot instances.
[Online]. Available: https://aws.amazon.com/cn/ec2/spot/pricing/.

[12] M. Chen, S. Huang, X. Fu, X. Liu, and J. He, Statistical Model Checking-Based Evaluation and Optimization for Cloud Workflow Resource Allocation, *IEEE Transactions on Cloud Computing*, 2016.

[13] G. Gallego and G. Ryzin, Optimal dynamic pricing of inventories with stochastic demand over finite horizons, *Management Science*, vol. 40, no. 8, pp. 999-1020, 1994.

[14] S. Karlin and C. Carr, Prices and optimal inventory policy, *Arrow Karlin and Scarf Studies in Applied Probability and Management Science*, pp. 159-172, 1962.

[15] P. Pfeiffer. Probability for applications, *Springer* , 2012.

[16] H. Yao, C. Bai, D. Zeng, Q. Liang, and Y. Fan, Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud, *Concurrency and Computation: Practice and Experience*, vol. 27, pp. 5780-5792, 2015.

[17] K. Li, Optimal load distribution for multiple heterogeneous blade servers in a cloud computing environment, *Journal of Grid Computing*, pp. 943-952, 2011.

[18] B. Chun and and D. Culler, User-centric performance analysis of market-based cluster batch schedulers, *International Symposium on Cluster Computing and the Grid*, 2002.

[19] K. Li, Optimal configuration of a multicore server processor for managing the power and performance tradeoff, *Journal of Supercomputing*, vol. 61, no. 1, pp. 189-214, 2012.

[20] L. Kleinrock, Queueing systems, Volume 1: Theory, *Wiley* , 1975.

[21] J. Little and S. Graves, Little's law, *International Series in Operations Research and Management Science*, pp. 81-100, 2008.

[22] W. Long, X. Liang, S. Cai, J. Jiao, and W. Zhang, A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems, *Neural Computing and Applications*, pp. 1-18, 2016.

[23] Y. Zheng and Z. Meng, A new augmented Lagrangian objective penalty function for constrained optimization problems, *Open Journal of Optimization*, vol. 6, pp. 39-46, 2017.

[24] B. Dandurandet, N. Boland, J. Christiansen, A. Eberhard, and F. Oliveira, A parallelizable augmented Lagrangian method applied to large-scale non-convex-constrained optimization problems,
[Online]. Available: https://arxiv.org/abs/1702.00526.

[25] Completing the square-wikipedia.
[Online]. Available: https://en.wikipedia.org/wiki/Completing_the_square.

[26] Y. Zhan, M. Ghamkhari, D. Xu, S. Ren, and H. Mohsenian-Rad, Extending demand response to tenants in cloud data centers via non-intrusive workload flexibility pricing, *IEEE Transactions on Smart Grid*, pp. 1-8, 2016.

[27] Browse real-time traffic to google products and services. [Online]. Available: http://www.google.com/transparencyreport/traffic/explorer.

[28] Real time prices-ameren.
[Online]. Available: https://www.ameren.com/RetailEnergy/RealTimePrices.

[29] Z. Yang and R. Peterson, Customer perceived value, satisfaction and loyalty: The role of switching costs, *Psychology and Marketing*, vol. 21 no. 10, pp. 799-822, 2004.