IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 4, 2020, accepted June 15, 2020, date of publication June 19, 2020, date of current version July 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003799

A Profit Maximization Scheme in Cloud Computing With Deadline Constraints

SIYI CHEN[1], SINING HUANG[1], QIANG LUO[2], AND JIALING ZHOU[1]
[1]School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China
[2]School of Civil Engineering, Guangzhou University, Guangzhou 510640, China

Corresponding authors: Siyi Chen (c.siyi@xtu.edu.cn) and Qiang Luo (luoq_yan@gzhu.edu.cn)

This work was supported in part by the National Key Research and Development Project under Grant 2018YFB142900, in part by the General Project of Education Department of Hunan Province under Grant 18C0091, and in part by the Guangzhou University Research Project under Grant YG2020004.

ABSTRACT Cloud computing has attracting more and more attention for its flexibility and economic benefits. To maintain the supply-demand relationship among different participants in cloud computing environment, the exchange of value is the inner drive. From the perspective of cloud service provider, its primary concern is to earn the profit, which can be obtained by finishing the tasks published from customers. In this paper, we consider each task consists of numbers of sub-tasks in the logical order, each sub-task corresponds to a type of service requests, which can be served in unique multi-server system. On this basis, we propose a profit maximization problem in the multistage multi-server queue systems, in which customers are served at more than one stage, arranged in a series structure. Moreover, a deadline constraint is taken into consideration, which demonstrates the maximum tolerance degree that the customers can wait. Therefore, how to configure the parameters in multistage multi-server queue systems to maximize profit on the premise of reducing the waiting times of customers is a critical issue for cloud service provider. To address this problem, we first discuss the probability distribution function of the waiting time for single multi-server system and multistage multi-server queue systems respectively, and then propose a profit maximization model under the deadline constraint. Due to the complexity of this model, the analytical solution can hardly be obtained, we study a heuristic method to search for the optimal solution. At last, a series of numerical simulations are implemented to describe the performance of the proposed profit maximization scheme, the results show that not only the profit can be maximized, but also the waiting time of customers have been reduced effectively.

INDEX TERMS Cloud computing, deadline, queueing model, multi-server system, profit maximization, waiting time.

I. INTRODUCTION

Cloud computing has contracting more and more attention in the past decade [1]. As a service related to information, software and internet, cloud computing integrates a large amount of resources and services, and delivers them on the internet. Customers obtain these resources and services on demand without considering the maintenance and management of the hardware [2]. Due to the excellent characteristics, customers can improve work efficiency and user experience, and reduce large capital outlays and human expenses [3]. However, the resources and services are not provided for free. In order to maintain the operation of cloud computing, cloud service providers will charge customers the necessary fees by adopting the pay-per-use pricing model [4]. Therefore, it is essential to understand the economics of cloud computing.

Generally, depending on the purposes of different participants, a cloud computing environment can be considered as a three-tier structure, which consists of infrastructure providers, cloud service providers and customers. The infrastructure providers maintain physical devices, and use them to construct dynamic resource pool by adopting virtualization technologies. Cloud service providers rent resources from infrastructure providers and pay the rental cost correspondingly, meanwhile, they build the cloud computing platform for providing services to customers. Customers search for

The associate editor coordinating the review of this manuscript and approving it for publication was Muhamamd Aleem.

the solutions on the platform according to their requirements, and charge for the provided services based on their quantities and qualities. As the connecting link between infrastructure providers and customers, cloud service providers are of great importance [5]. Moreover, profits are the foundation of the normal operation of the cloud computing platform, which consist of the revenues from customers and the costs to infrastructure providers [6]. In this paper, we focus on the research of profit maximization of cloud service providers.

The factors which can affect the profit of cloud service providers have been investigated in numerous literatures, such as market demand, configuration of the parameters in cloud computing platform, pricing model and so on. Consider the customer-oriented service demand is the fundamental to the management mechanism of cloud computing, besides, the quality of service and the price of service are the most concern to customers, consequently, the optimal configuration of the parameters in cloud computing platform and pricing model are the most important among all of these factors [7]. However, high quality of service always brings high cost to cloud service providers, which will enforce them to raise the price of service to earn profit. On the contrary, low price of the service will cause a decline in the quality of service. Therefore, it is important for cloud service providers to address the trade-off between increasing the quality of service and decreasing the price of service, so as to maximize the profit. Cao *et al.* [8] studied a problem of optimal multi-server configuration for profit maximization in cloud computing environment, the number and the execution speed of servers are indicated as the basic features in determining the configuration of a multi-server system. Ghamkhari and Mohsenian-Rad [9] analyzed the SLA between cloud service providers and customers, and pointed out that energy expenditure is the major consumption in managing the cloud computing platform. Consider the subjective willingness of customers in purchasing cloud services and the corresponding influence on the profit of cloud service providers, Cong *et al.* [10] introduced the concept of user perceived value, and proposed a profit maximization scheme based on the dynamic pricing model to optimize the profit by configuring the parameters in multi-server system under the constraint of service-level agreement. Mei *et al.* [11] introduced the definition of customer satisfaction in economics, demonstrated that how the configuration of cloud computing platform affects the quality of service and customer satisfaction, and how the customer satisfaction further affects the profit. However, these methods rarely focused on the profit maximization scheme in single multi-server system, which can only be adopted to serve one type of service requests [12]. For most cases, when the customer publishes a task, it can always be separated into numbers of sub-tasks in the logical order, each of which corresponds to a type of service requests. Only if the front sub-tasks are completed, the latter sub-tasks can be handled on the basis of the corresponding results obtained in the front sub-tasks. Therefore, to fulfill the task published by a customer, the multistage multi-server queue systems should be analyzed.

In this paper, we study a profit maximization scheme by configuring the parameters in multistage multi-server queue systems arranged in a series structure, each system is treated as an M/M/m queueing model. At each stage, one type of service requests, which corresponds to a sub-task published by a customer, can be served in unique multi-server system. For the customers with limited patience, namely, the total waiting time that they are willing to spend on multistage multi-server queue systems can not exceed the deadline [13], cloud service providers should configure the parameters in cloud computing platform to satisfy the demands of customers as much as possible, and so as to obtain more revenues. However, the cost will also increase because of the growing energy expenditure and rental cost. Therefore, how to optimal configure the parameters to maximize profit with deadline constraint is an important problem. This problem consists of three sub problems. Firstly, how to model the multistage multi-server queue systems arranged in a series structure. Secondly, how to determine the total waiting times that a customer spend on the multistage multi-server queue systems. Lastly, how to realize the maximization of profit by configuring cloud computing platform under deadline constraint.

The main contributions of this paper are summarized as follows.

- Consider the multistage multi-server queue systems arranged in a series structure, analyze the revenue and cost model of cloud service providers in each multi-server system, and build a profit maximization model.
- Based on the maximum tolerance degree that a customer can wait, define a profit maximization problem under the deadline constraint, and describe the heuristic algorithm to solve the problem, so as to realize the maximization of profit and the decline in the waiting time of customer simultaneously.
- Perform a series of numerical simulations to investigate the performance of the proposed algorithm, and investigate the variations of profit and percentage of executed service requests with an increasing level of deadline and the arrival rate of service requests.

The rest of this paper is organized as follows. Section II reviews the related work on profit maximization scheme. Section III presents the three-tier cloud structure, the multistage multi-server queue system model, the revenue model, the cost model, and the profit model. Section IV studies the profit maximization scheme in each multi-server system step by step. Section V describes the performance of the proposed algorithm. At last, Section VI concludes the work.

II. RELATED WORK

In this section, we first review the literatures concerning the optimal configuration of the parameters in cloud computing platform under deadline constraint, and then the profit maximization problem in multi-server systems.

Deadline constraint reflects the maximum length of time that a service request can wait when it has not been served yet, which is one of the main factors in configuring the cloud computing platform [14]. Chang *et al.* [15] adopted rough set theory to estimate the execution time of service request, so as to satisfy the deadline constraint on certain virtual machine (VM) with given resource. Deldari *et al.* [16] divided the workflow into a number of clusters, and then adopted an extendable scoring approach to choose best cluster combinations to minimize the execution cost when considering the deadline constraint. Zou *et al.* [17] considered a deadline satisfaction problem in line-of-balance (LOB) scheduling, and proposed a biobjective optimization model to address the trade-off between minimizing the total number of crews and maximizing work continuity. Li *et al.* [18] presented a slot-based data structure to organize available resources of multiprocessor systems, so that they can be allocated to parallel advance reservation jobs with deadline constraint. On the basis, Li *et al.* [19] considered the deadline and time slot availability constraint both for workflow scheduling problem, for the reason that the two constraints are crucial for saving the costs in cloud computing with limited service capacities. Recently, Canon*et al.* [20] focused on the scheduling problem within a given budget and deadline constraint to maximize the expected number of tasks in cloud computing platform. Moreover, consider the multistage multi-server queue systems discussed in this paper, each system only devotes to serve one type of service requests (sub-task), then customers have to be served at all of the stages successively to fulfill their tasks. Therefore, compare to the maximum length of time that a service request can wait, we are more concerned about the maximum length of time that a customer can wait, which can be separated into the waiting time of each service request in the corresponding multi-server queue system.

When considering the profit maximization problem in multiple multi-server systems, the related researches mainly focused on the multi-server systems with a parallel structure. For example, Lan [21] considered a production designing and scheduling problem in a manufacturing system with multiple parallel production lines, and proposed a profit maximization scheme by configuring some parameters appropriately, such as the suggested production rate, the production time interval, and so on. Su [22] addressed the identical parallel machine scheduling problem with job deadlines and machine eligibility constraints to minimize total job completion time. Li *et al.* [12] built a fund-constrained profit maximization model for a group of n heterogenous multi-server systems, and then discussed a parameter configuration and a fund allocation problem to achieve the maximization of profit. Since the multiple multi-server systems are arranged in the parallel structure in these methods, then multiple types of service requests can be served simultaneously. However, they neglected the inner-relationship among all the service requests (sub-tasks), cause that some service requests (sub-tasks) must be served successively to fulfill the task published by the customer. Namely, only if the front sub-tasks are completed, the latter sub-tasks can be handled on the basis of the corresponding results obtained in the front sub-tasks. Therefore, it is essential to study a profit maximization problem in the multistage multi-server queue systems arranged in a series structure.

While considering the queue systems with multiple stages, Toktaş-Palut [23] investigated a two-stage supply chain consisting of multiple suppliers at the first stage and a manufacturer at the second stage, each supplier was modeled as a M/M/1 queue and the manufacturer was modeled as a GI/M/1 queue. During the investigation, the interarrival time of manufacturer was proven to be equal to the interdeparture time of supplier, which reflected the inner-relationship of the first stage and second stage. However, they only considered the case that only one of two stages can perform its tasks in a certain period of time. Thangaraj and Vanitha [24] analyzed two stages of heterogeneous service with different service time distributions subject to random breakdowns and compulsory server vacations with general vacation periods. The average number of customers in the queue and the average waiting time were discussed. Ramasamy *et al.* [25] presented the steady state analysis of a heterogeneous Geo/G/2 queuing system, a serial queue discipline and a parallel queue discipline were employed for service respectively. By adopting such alternative queue disciplines, some violations of first-come-first-serve (FCFS) principle occurred because of the heterogeneity of servers are minimized. Sundari and Srinivasan *et al.* [26] considered a three-stage M/G/1 Bernoulli feedback queue with multiple server vacation, the customers who arrived the queue system will undergo three stages of service, each stage was executed by a single server. Based on such method, the expected number of customers in single queue as well as in the system are deduced, and so as to the mean waiting time in single queue as well as in the system respectively. Ajiboye and Saminu [27] studied a multistage queue system with a certain number of independent parallel servers and multiple queues at all or some of the stages, and provided an effective method to manage the queues for maximizing customer satisfaction with no additional cost. All of these methods mainly focus on solving the problems in the queue system with single server in a stage, but rarely focus on the case with multiple servers in a stage, which deserves to be further discussed.

III. THE MODELS

A. CLOUD COMPUTING STRUCTURE

The typical three-tier cloud computing structure is shown in Figure.1. To study the supply-demand relationship of services and applications in cloud computing environment, the behaviors and characteristics of infrastructure providers, cloud service providers and customers are always taken into consideration.

For infrastructure providers, they adopt virtualization technology to congregate various of IT resources (computation, network, storage, etc.), and provide them to the remote

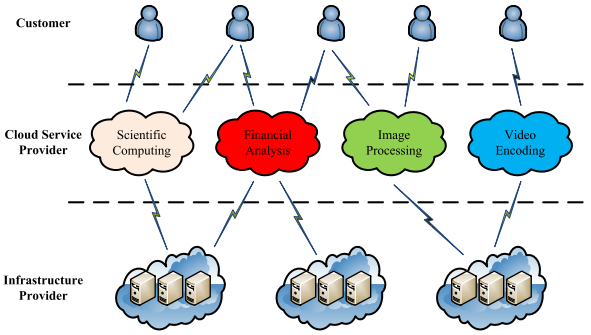


FIGURE 1. The three-tier cloud structure.

Internet customers on demand. Moreover, such IT resources are scalability, so that they can be adjusted according to the requirements of customers. A typical case is the number and execution speed of servers, which are variable for different applications.

For cloud service providers, they devote to establish the channel between infrastructure providers and customers, so that the customers have no need to pay attention to the specific implementation details of service requests. In practice, cloud service providers rent resources from infrastructure providers, and build the cloud computing platform to provide services to customers.

For customers, they submit the service requests to cloud service providers and pay for the provided service according to a specified service-level agreement. In this case, some pricing model have been developed, such as flat rate pricing strategy [28], usage-based pricing strategy [29], dynamic pricing strategy [30] and so on.

B. MODELING MULTISTAGE MULTI-SERVER QUEUE SYSTEMS

Generally, when the customer publishes a task, it can always be separated into numbers of sub-tasks, and the order of execution of these sub-tasks should follow successive logical relationship. In this paper, we suppose each sub-task corresponds to a type of service requests, which can be served in unique multi-server system. On this basis, we consider a cloud computing platform as the multistage multi-server queue systems consist of n M/M/m queueing models $S_1, S_2, \cdots, S_n$ arranged in a series structure, which is shown in Figure.2. For each multi-server system S_i, it has m_i identical servers with speed s_i, where $i = 1, 2, \cdots, n$. According to Figure.2, once the customer publishes a task, the first sub-task (or the first type of service requests) will be served in the first multi-server system immediately when some servers are available. When the first stage is done, the multi-server systems after the first one will serve the latter sub-tasks (service requests) in the following stages sequentially.

Without loss of generality, we consider the multistage multi-server queue systems as a simplified form, in which such systems have only two stages, namely, $n = 2$. Then,

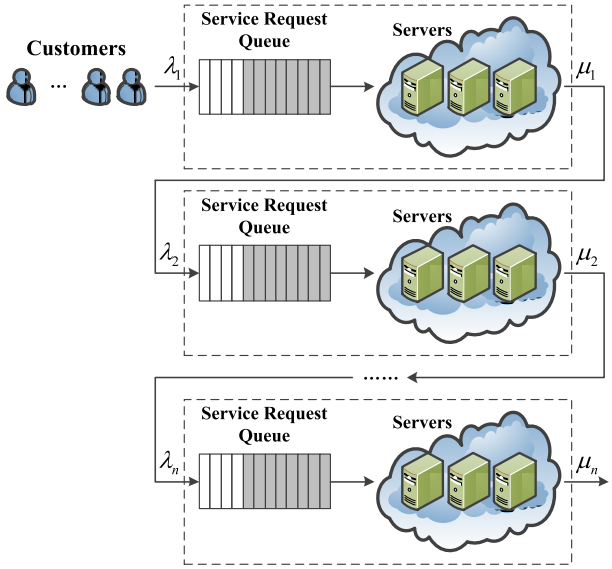


FIGURE 2. Multistage multi-server queue systems.

we will find that such simplified form can be easily extended to the general cases, which is described in Section IV-C.

For the multi-server system S_i in Figure.2, when a customer arrives the queue with the specific service requests (sub-task), the time it takes him to arrive is a random variable with an independent and identically distributed (i.i.d.) exponential distribution whose mean is $1/\lambda_i$ sec. In other words, the service requests follow a Poisson process with arrival rate λ_i. Due to the limited servers in each multi-server system, the incoming service requests may not be served immediately. In this paper, we assume that customers are impatient. When their service requests cannot be processed immediately after they arrive, they will be placed in the unlimited waiting queues which are maintained by the multi-server systems. However, once the total waiting times of customers spent in multistage multi-server queue systems exceed the deadline D, they will depart from the queues forever even when their service requests have not been served yet. The task execution requirements are exponential random variables r with mean $\bar{r}$, which represent the number of instructions to be processed. Then the execution times can also be thought as exponential random variables $t_i = r/s_i$ with mean $\bar{t}_i = \bar{r}/s_i$. Therefore, the service rate is $\mu_i^s = 1/\bar{t}_i = s_i/\bar{r}$ for the system with only one service requests being served by single server. While for the multi-server system, if the number of incoming service requests is less than the number of servers, they will be served immediately, otherwise, the execution of part of the service requests has to be delayed due to the limited number of servers. In this case, the service rate μ_i can be represented as follow.

$$\mu_i = \begin{cases} k_i \mu_i^s, & k_i = 1, 2, \cdots, m_i \\ m_i \mu_i^s, & k_i = m_i, m_i + 1, \cdots \end{cases} \tag{1}$$

Based on the arrival rate λ_i and service rate μ_i^s, we have the utilization factor $\rho_i = \lambda_i/(m_i \mu_i^s) = \lambda_i \bar{r}/(m_i s_i)$.

Denote p_{k_i} as the probability that there are k_i service requests (waiting or being processed) in each M/M/m queueing system S_i. Then, we can obtain

$$p_{k_i} = \begin{cases} p_{0,i}\dfrac{(m_i\rho_i)^{k_i}}{k_i!}, & k_i < m_i \\ p_{0,i}\dfrac{m_i^{m_i}\rho_i^{k_i}}{m_i!}, & k_i \geq m_i \end{cases} \tag{2}$$

where

$$p_{0,i} = \left(\sum_{k_i=0}^{m_i-1}\frac{(m_i\rho_i)^{k_i}}{k_i!} + \frac{(m_i\rho_i)^{m_i}}{m_i!}\frac{1}{1-\rho_i}\right)^{-1}$$

Since the two-stage multi-server systems are arranged in a series structure, only if the first type of service requests is served in the front stage, the customer can send the second type of service requests to the multi-server system in the latter stage, then we can find that the interdeparture times of the first multi-server system are equal to the interarrival times of the second multi-server system [23]. Therefore, it is reasonable to think that the arrival rate of service requests in the latter stage is equal to the average service rate of service requests in the front stage, namely, $\lambda_2 = \bar{k}\mu_1^s$, where $\bar{k}$ is the average number of service requests in execution in the first stage per unit of time. According to Eq.(1), the service rate is varying with the number of service requests, which follows the probability distribution as shown in Eq.(2). On this basis, we can describe the average service rate as an expectation form, which is shown as follow.

$$\begin{aligned}\bar{k}\mu_1^s &= \sum_{k_1=0}^{m_1-1}\left(p_{k_1}\cdot k_1\mu_1^s\right) + \sum_{k_1=m_1}^{\infty}\left(p_{k_1}\cdot m_1\mu_1^s\right) \\ &= p_{0,1}\mu_1^s\left[\sum_{k_1=0}^{m_1-1}\frac{(m_1\rho_1)^{k_1}}{(k_1-1)!} + \sum_{k_1=m_1}^{\infty}\frac{m_1^{m_1}\rho_1^{k_1}}{(m_1-1)!}\right] \\ &= p_{0,1}\rho_1 m_1\mu_1^s\left[\sum_{k_1=0}^{m_1-2}\frac{(m_1\rho_1)^{k_1}}{k_1!}\right. \\ &\quad \left. + \frac{(m_1\rho_1)^{m_1-1}}{(m_1-1)!} + \frac{m_1^{m_1-1}}{(m_1-1)!}\sum_{k_1=m_1+1}^{\infty}\rho_1^{k_1-1}\right] \\ &= p_{0,1}\rho_1 m_1\mu_1^s\left[\sum_{k_1=0}^{m_1-1}\frac{(m_1\rho_1)^{k_1}}{k_1!} + \frac{m_1}{m_1}\frac{m_1^{m_1-1}}{(m_1-1)!}\frac{\rho_1^{m_1}}{1-\rho_1}\right] \\ &= \rho_1 m_1\mu_1^s \end{aligned} \tag{3}$$

Actually, it can be easily found that the average service rate is also equal to the arrival rate in the same stage, namely, $\lambda_1 = \bar{k}\mu_1^s$. This has an intuitive interpretation. Given the condition $\rho_1 < 1$ to ensure the ergodicity of queue system, the incoming service requests to the first multi-server system can be served without waiting in the long run. However, such point is not correct within small time intervals, cause the incoming service request is essentially a kind of random flow, which may result in the occasional bursts of traffic to temporarily overwhelm the servers([31], p. 99). On this basis, when all servers in a multi-server system are occupied by the executed service requests, then the newly arrived service requests must wait in the waiting queue. In this case, we denote its probability as follow.

$$P_{q_i} = \sum_{k_i=m_i}^{\infty} p_{k_i} = \frac{p_{m_i}}{1-\rho_i} = p_{0,i}\frac{(m_i\rho_i)^{m_i}}{m_i!}\frac{1}{1-\rho_i} \tag{4}$$

Let W_i denote the waiting time of the ith type of service request, the corresponding probability distribution function (pdf) can be described as follow [8].

$$f_{W_i}(t) = \left(1-P_{q_i}\right)u(t) + m_i\mu_i^s p_{m_i}e^{-(1-\rho_i)m_i\mu_i^s t} \tag{5}$$

where $u(t)$ is an unit impulse function, which is defined as

$$u_z(t) = \begin{cases} z, & 0 \leq t \leq \dfrac{1}{z} \\ 0, & t > \dfrac{1}{z} \end{cases}$$

Let $z \to \infty$, then we have

$$u(t) = \lim_{z\to\infty} u_z(t) \tag{6}$$

The function $u_z(t)$ has the following properties

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

and

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

Theorem 1: Suppose waiting times of service requests spent in the first and second multi-server system are W_1 and W_2 respectively, if and only if the condition $m_2\mu_2^s < m_1\mu_1^s$ is satisfied, then the pdf of the total waiting time $W = W_1 + W_2$ of the customer is

$$f_W(t) = Ae^{-(1-\rho_1)m_1\mu_1^s t} + Be^{-(1-\rho_2)m_2\mu_2^s t} \tag{7}$$

where

$$A = \left(1-P_{q_2}\right)m_1\mu_1^s p_{m_1}$$

$$B = \left(1-P_{q_1}\right)m_2\mu_2^s p_{m_2} + \frac{\prod_{i=1}^{2}\left(m_i\mu_i^s p_{m_i}\right)}{(1-\rho_1)m_1\mu_1^s - (1-\rho_2)m_2\mu_2^s}$$

Proof: The proof is given in the Appendix A. □

C. REVENUE MODELING

Customers enjoy the services provided by cloud service providers on demand, and certainly they will pay for them. To study the actual service charge to customer, a Service-Level Agreement (SLA) is adopted, which clearly demonstrates the relationship between Quality of Service (QoS) and the corresponding charge. In this paper, we choose waiting time to represent the difference in QoS, for it is intuitive and can be easily obtained. The service charge functions

$R_i (i = 1, 2)$ for a service request in the first and second multi-server system are defined as follow.

$$R_1(r, W_1) = \begin{cases} a_1 r, & 0 \le W_1 \le D \\ 0, & W_1 > D \end{cases} \tag{8}$$

$$R_2(r, W_1, W_2) = \begin{cases} a_2 r, & 0 \le W_2 \le D - W_1 \\ 0, & W_2 > D - W_1 \end{cases} \tag{9}$$

where a_1, a_2 are constants, which represent the service charges per unit of service, and D is the maximum tolerable time that the service requests can wait. In this paper, we assume that the cloud service providers charge customers with a constant when the waiting time does not exceed the maximum value. For the given two-stage multi-server systems, such assumption can be separated into three cases. Firstly, if the waiting time of the first type of service requests spend in the front stage exceed the deadline, the customers will depart from the two-stage multi-server systems even when their service requests have not been served yet, and they certainly should not pay for them. Secondly, if the first type of service requests is served within the deadline, while the total waiting time exceeds, the customers will depart from the second multi-server system even when the second type of service requests has not been served yet. In this case, they will only pay for the first type of service requests. Lastly, if the total waiting time does not exceed the deadline, the tasks published by customers are fulfilled successfully, then the customers will pay for both types of service requests.

Based on Eq.(8) and (9), the expected charge of a service request in multi-server system S_1 and S_2 can be obtained by the following theorem.

Theorem 2: The expected charge to a service request in multi-server system S_1 and S_2 are Eq.(10) *and* (11) *respectively.*

$$\bar{R}_1 = a_1 \bar{r} \left(1 - \frac{p_{m_1}}{1 - \rho_1} e^{-m_1 \mu_1^s (1-\rho_1) D}\right) \tag{10}$$

$$\bar{R}_2 = a_2 \bar{r} \left(A' \left(1 - e^{-(1-\rho_1) m_1 \mu_1^s D}\right) + B' \left(1 - e^{-(1-\rho_2) m_2 \mu_2^s D}\right)\right) \tag{11}$$

where

$$A' = \frac{A}{(1-\rho_1) m_1 \mu_1^s}, \quad B' = \frac{B}{(1-\rho_2) m_2 \mu_2^s}$$

Proof: The proof is given in the Appendix B. □

For convenience, we rewrite Eq.(10) and (11) as $\bar{R}_1 = F_{W_1}(D)\, a_1 \bar{r}$ and $\bar{R}_2 = F_W(D)\, a_2 \bar{r}$ respectively, where $F_{W_1}(D)$ and $F_W(D)$ represent the percentage of the service requests which can be served within the first and second stage respectively. Notice that, due to the deadline constraint, only part of the first type of service requests can be served in the front stage, which will result in a decline in the arrival rate of service requests to the latter stage. Therefore, when the service requests enter multi-server system S_1, the arrival rate is λ_1, but when the service requests enter multi-server system S_2, the arrival rate becomes $F_{W_1}(D)\lambda_2 = F_{W_1}(D) \bar{k} \mu_1^s = F_{W_1}(D) m_1 \rho_1 \mu_1^s$, for the reason that only $F_{W_1}(D)$ percent of incoming service requests can be handled in multi-server system S_1 before the deadline D, while the rest will depart without being served. Therefore, the total revenue obtained by cloud service providers in multi-server system S_1 and S_2 can be described as

$$\varepsilon_1 = \lambda_1 \bar{R}_1 = \lambda_1 F_{W_1}(D) a_1 \bar{r} \tag{12}$$

$$\begin{aligned} \varepsilon_2 &= F_{W_1}(D) \lambda_2 \bar{R}_2 \\ &= F_{W_1}(D) m_1 \rho_1 \mu_1^s F_W(D) a_2 \bar{r} \\ &= \lambda_1 F_{W_1}(D) F_W(D) a_2 \bar{r} \end{aligned} \tag{13}$$

D. COST MODELING

The costs to service providers consist of two major parts, i.e., the cost of infrastructure renting and utility cost of energy consumption. Infrastructure providers maintain a large amount of servers for lease, cloud service providers rent them according to the requirements and pay the corresponding rental costs. Assuming that the rental price of one server per unit of time is β, one can obtain the server rental price for a multi-server system with m_i servers is $m_i \beta$.

As another part of the costs to service providers, the utility cost of energy consumption is composed of electricity price and the amount of energy consumption. In this paper, the following dynamic power model is adopted, which had been discussed in many literatures [8], [32]–[34].

$$P_d = N_{sw} C_L V^2 f \tag{14}$$

where N_{sw} is the average gate switching factor at each clock cycle, C_L is the loading capacitance, V is the supply voltage, and f is the clock frequency. In the ideal case, the relationship between supply voltage V and clock frequency f can be described as $V \propto f^{\phi}$ for some constant $0 < \phi \le 1$. The execution speed of server s_i is linearly proportional to the clock frequency f, namely, $s_i \propto f$. Therefore, the dynamic power model can be transformed into $P_d \propto N_{sw} C_L s_i^{2\phi+1}$. For the sake of simplicity, we assume that $P_d = b N_{sw} C_L s_i^{2\phi+1} = \xi s_i^{\alpha}$, where $\xi = b N_{sw} C_L$, $\alpha = 2\phi + 1$ and b is a constant. In this paper, we set $N_{sw} C_L = 7$, $b = 1.3456$ and $\phi = 0.5$. Therefore, we obtain $\alpha = 2$ and $\xi = 9.4192$.

Apart from dynamic power consumption, it is reasonable to think that some amount of static power are consumed by the servers when they are idle. In this case, the average amount of energy consumption per unit of time can be described as $P = \xi s_i^{\alpha} + P^*$, where P^* is the static power consumption.

Notice that, the necessary and sufficient condition for ergodicity in the M/M/m queueing system is $\rho_i < 1$ ([31], p. 95). However, $P = \xi s_i^{\alpha} + P^*$ implies that $\rho_i = 1$. Therefore, the average amount of energy consumption per unit of time can be further described as $P = \rho_i \xi s_i^{\alpha} + P^*$. Assuming that the price of energy is δ per Watt, then the total cost of the multi-server system per unit of time can be described as

$$C_i = m_i \left(\beta + \delta \left(\rho_i \xi s_i^{\alpha} + P^*\right)\right) \tag{15}$$

According to the analysis in Section III-C, the customer whose waiting time exceeds the deadline will depart from the

multi-server systems, which causes a decrease in the revenue of cloud service providers. However, since only $F_{W_1}(D)$ percent of the first type of service requests are executed by S_1, and $F_W(D)$ percent of the second type of service requests are further executed by S_2, the utilization factor ρ_1 and ρ_2 will reduce to $\rho_1 F_{W_1}(D)$ and $\rho_2 F_W(D)$, respectively, [35]. Therefore, the total cost of the multi-server systems can be further described as

$$C_1 = m_1\left(\beta + \delta\left(\rho_1 F_{W_1}(D)\xi s_1^{\alpha} + P^*\right)\right) \tag{16}$$
$$C_2 = m_2\left(\beta + \delta\left(\rho_2 F_W(D)\xi s_2^{\alpha} + P^*\right)\right) \tag{17}$$

E. PROBLEM DESCRIPTION

Cloud service providers rent servers from infrastructure providers and pay the costs, meanwhile, they provide services to customers on demand and obtain the revenues. As can be seen from the above analysis, the maximum waiting time that the customer can tolerate have an impact both on the cost model and revenue model of cloud service providers in each multi-server system. Hence, for a cloud computing platform consists of multistage multi-server queue systems, it is essential to study an appropriate method to maximize the total profits of cloud service providers under the deadline constraint.

According to the structure of multistage multi-server queue systems as shown in Figure.2, the total profits of cloud service providers consist of each part obtained in multi-server system S_1 and S_2 respectively, which is shown in Eq.(18). In this paper, we devote to optimize the number of rental servers m_i and the execution speed s_i to obtain the optimal profit.

$$G(m_1, m_2, s_1, s_2) = G_1(m_1, s_1) + G_2(m_1, m_2, s_1, s_2) \tag{18}$$

Notice that, G_1 is only determined by the characteristics of S_1 itself, while G_2 is determined by the characteristics of S_1 and S_2 together. It is obvious, since the execution of the second type of service requests in the latter stage lags behind the execution of the first type of service requests in the front stage, then the parameters in the second multi-server system are irrelevant to the profit obtained in the first multi-server system. However, such point is not correct contrarily, because the execution of the second type of service requests is affected by the waiting time of the first type of service requests. The more the waiting time spend in the front stage, the less the waiting time can be spent in the latter stage, which will cause a decline in the profit obtained in the second multi-server system, and vice versa. Above all, the total profit of a cloud service provider can be described as follow. In this paper, we set $\bar{r} = 1, D = 5, a_1 = a_2 = 15, P^* = 3, \alpha = 2$, $\beta = 1.5, \xi = 9.4192$ and $\delta = 0.3$.

$$\begin{aligned} &G(m_1, m_2, s_1, s_2) \\ &= \varepsilon_1 - C_1 + \varepsilon_2 - C_2 \\ &= \lambda_1 F_{W_1}(D) a_1 \bar{r} - m_1\left(\beta + \delta\left(\rho_1 F_{W_1}(D)\xi s_1^{\alpha} + P^*\right)\right) \\ &\quad + \lambda_1 F_{W_1}(D) F_W(D) a_2 \bar{r} - m_2\left(\beta + \delta\left(\rho_2 F_W(D)\xi s_2^{\alpha} + P^*\right)\right) \end{aligned} \tag{19}$$

IV. A HEURISTIC ALGORITHM

In this section, we propose a heuristic algorithm to find the optimal combination of m_1, s_1 and m_2, s_2, so as to achieve the profit maximization step by step. First, the relationship among the profit G_1 and m_1 as well as s_1 is analyzed in multi-server system S_1, and the gradient descent algorithm is adopted to configure the optimal server parameters to obtain the optimal profit. Second, the relationship among the profit G_2 and m_2 as well as s_2 is analyzed on the basis of the server parameters obtained in the front stage, and an optimal model with constraint is built to maximize G_2 and $F_W(D)$ simultaneously.

A. MAXIMIZE PROFIT IN THE FIRST MULTI-SERVER SYSTEM

1) OPTIMAL SIZE

To obtain the maximum profit in the S_1, the influence of the number of servers m_1 on G_1 is first discussed. Since G_1 is a function of m_1 and s_1, we adopt partial derivative of G_1 with respect to m_1 to find the optimal m_1, which can be described as follow.

$$\frac{\partial G_1(m_1, s_1)}{\partial m_1} = \frac{\partial \varepsilon_1}{\partial m_1} - \frac{\partial C_1}{\partial m_1} = 0 \tag{20}$$

where

$$\frac{\partial \varepsilon_1}{\partial m_1} = \lambda_1 a_1 \bar{r} \frac{\partial F_{W_1}(D)}{\partial m_1}$$

and

$$\frac{\partial C_1}{\partial m_1} = \beta + \delta P^* + \lambda_1 \bar{r} \delta \xi s_1^{\alpha - 1} \frac{\partial F_{W_1}(D)}{\partial m_1}$$

For convenience, we set $\sum_{k_i=0}^{m_i-1} \frac{(m_i\rho_i)^{k_i}}{k_i!} \approx e^{m_i\rho_i}$ and $m_i! \approx \sqrt{2\pi m_i}\left(\frac{m_i}{e}\right)^{m_i}$ [36], then $F_{W_1}(D)$ can be rewrite as

$$\begin{aligned} F_{W_1}(D) &\approx 1 - \frac{e^{-m_1\mu_1^s(1-\rho_1)D}}{\sqrt{2\pi m_1}(1-\rho_1)\left(\frac{e^{\rho_1}}{e\rho_1}\right)^{m_1} + 1} \\ &= 1 - \frac{\sigma_1}{1+\sigma_2} \end{aligned} \tag{21}$$

where $\sigma_1 = e^{-m_1\mu_1^s(1-\rho_1)D}$, $\sigma_2 = \sqrt{2\pi m_1}(1-\rho_1)\varphi$ and $\varphi_1 = \left(\frac{e^{\rho_1}}{e\rho_1}\right)^{m_1}$. Since

$$\ln\varphi_1 = m_1 \ln\left(e^{\rho_1}/e\rho_1\right) = m_1(\rho_1 - \ln\rho_1 - 1) \tag{22}$$

Then, we have

$$\frac{\partial \ln\varphi_1}{\partial m_1} = \frac{1}{\varphi_1}\frac{\partial\varphi_1}{\partial m_1} = (\rho_1 - \ln\rho_1 - 1) + m_1\left(1 - \frac{1}{\rho_1}\right)\frac{\partial\rho_1}{\partial m_1} \tag{23}$$

Simplify the equation, we have $\frac{\partial \ln\varphi_1}{\partial m_1} = -\varphi_1 \ln\rho_1$. On this basis, the partial derivative of σ_1 and σ_2 to m_1 can be described as

$$\begin{aligned} \frac{\partial\sigma_1}{\partial m_1} &= -\mu_1^s D\sigma_1 \\ \frac{\partial\sigma_2}{\partial m_1} &= \sqrt{2\pi m_1}\varphi_1\left[\frac{1}{2m_1}(1+\rho_1) - \ln\rho_1(1-\rho_1)\right] \end{aligned} \tag{24}$$

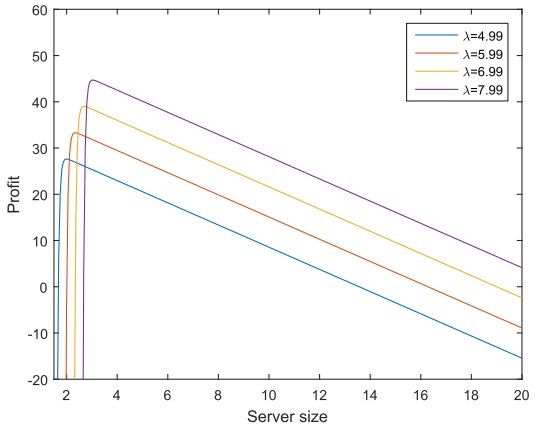


FIGURE 3. Profit G_1 versus m_1 and λ_1.

Then, we get

$$\frac{\partial F_{W_1}(D)}{\partial m_1} = \frac{\sigma_1 \frac{\partial \sigma_2}{\partial m_1} - (\sigma_2+1)\frac{\partial \sigma_1}{\partial m_1}}{(\sigma_2+1)^2} = \frac{\mu_1^s D \sigma_1}{\sigma_2+1} + \sqrt{\frac{\pi}{2m_1}}\sigma_1\varphi_1(1+\rho_1) - \sqrt{2\pi m_1}\sigma_1\varphi_1 \ln\rho_1(1-\rho_1) \quad (25)$$

Apparently, the analytical solution of Eq.(20) cannot be calculated. By drawing the curves of profit G_1 as a function of m_1 with fixed s_1 and λ_1 in Figure.3, we find that $\partial G_1/\partial m_1$ is a decreasing function of m_1. Therefore, we can adopt the standard bisection method to obtain the optimal m_1 such that G_1 is maximized.

Using the analytical method, the optimal m_1 can be found to satisfy $\partial G_1/\partial m_1 = 0$. For $\lambda_1 = 4.99, 5.99, 6.99, 7.99$, the optimal value of m_1 is $1.9911, 2.3375, 2.6818, 3.0247$ and the corresponding profit G_1 is $27.621, 33.312, 39.007, 44.707$ respectively.

As can be seen from Figure.3, cloud service providers can only obtain extremely low or even negative profits in S_1 when m_1 is low, this is because the waiting time of service requests are extremely long, which will result in the fact that only few service requests can be served within the deadline D, namely, $F_{W_1}(D) \ll 1$. As m_1 increases, the growing number of servers allow more and more service requests to be served within the deadline, then the revenues and profits are increased. Furthermore, the revenues reach the maximum value when $F_{W_1}(D)$ is equal to 1, but the costs will continue to grow as m_1 further increases, which will result in a decline in the profits instead. This is because the number of servers exceeds the maximum number required to execute the service requests.

2) OPTIMAL SPEED

Considering the influence of the execution speed s_1 on G_1, to maximize the profit, we adopt the partial derivative of G_1 with respect to s_1, which is described as follow.

$$\frac{\partial G_1(m_1,s_1)}{\partial s_1} = \frac{\partial \varepsilon_1}{\partial s_1} - \frac{\partial C_1}{\partial s_1} = 0 \quad (26)$$

where

$$\frac{\partial \varepsilon_1}{\partial s_1} = \lambda_1 a_1 \bar{r}\frac{\partial F_{W_1}(D)}{\partial s_1}$$

and

$$\frac{\partial C_1}{\partial s_1} = \lambda_1 \bar{r}\delta\xi \frac{\partial s_1^{\alpha-1} F_{W_1}(D)}{\partial s_1} = \lambda_1 \bar{r}\delta\xi\left((\alpha-1)s_1^{\alpha-2}F_{W_1}(D) + s_1^{\alpha-1}\frac{\partial F_{W_1}(D)}{\partial s_1}\right)$$

Based on Eq.(22), we have

$$\frac{\partial \ln\varphi_1}{\partial s_1} = \frac{1}{\varphi_1}\frac{\partial \varphi_1}{\partial s_1} = m_1\left(1-\frac{1}{\rho_1}\right)\frac{\partial \rho_1}{\partial s_1} \quad (27)$$

Since

$$\frac{\partial \rho_1}{\partial s_1} = -\frac{\lambda_1 \bar{r}}{m_1 s_1^2} = -\frac{\rho_1}{s_1} \quad (28)$$

Substituting Eq.(28) into Eq.(27), we get

$$\frac{\partial \varphi_1}{\partial s_1} = \frac{m_1}{s_1}(1-\rho_1)\varphi_1 \quad (29)$$

On this basis, the partial derivative of σ_1 and σ_2 to s_1 can be described as

$$\frac{\partial \sigma_1}{\partial s_1} = -D\sigma_1\frac{m_1}{\bar{r}}$$
$$\frac{\partial \sigma_2}{\partial s_1} = \frac{\sqrt{2\pi m_1}\varphi_1}{s_1}\left(\rho_1 + m_1(1-\rho_1)^2\right) \quad (30)$$

Then, we get

$$\frac{\partial F_{W_1}(D)}{\partial s_1} = \frac{\sigma_1\frac{\partial \sigma_2}{\partial s_1} - (\sigma_2+1)\frac{\partial \sigma_1}{\partial s_1}}{(\sigma_2+1)^2} = \frac{D\sigma_1 m_1}{(\sigma_2+1)\bar{r}} + \frac{\sqrt{2\pi m_1}\varphi_1}{(\sigma_2+1)^2 s_1}\left(\rho_1 + m_1(1-\rho_1)^2\right) \quad (31)$$

Similarly, the analytical solution of Eq.(26) cannot be calculated. By drawing the curves of profit G_1 as a function of s_1 with fixed m_1 and λ_1 in Figure.4, we find that $\partial G_1/\partial s_1$ is a decreasing function of s_1. Therefore, we can adopt the standard bisection method to obtain the optimal s_1 such that G_1 is maximized.

Using the analytical method, the optimal s_1 can be found to satisfy $\partial G_1/\partial s_1 = 0$. For $\lambda_1 = 4.99, 5.99, 6.99, 7.99$, the optimal value of s_1 is $0.20144, 0.23578, 0.26987, 0.30379$ and the corresponding profit G_1 is $-0.06463, 13.766, 27.408, 40.862$ respectively.

As can be seen from Figure.4, cloud service providers can only obtain extremely low or even negative profits in S_1 when s_1 is low, this is because only few service requests can be served per unit of time, while the rest depart from the system due to the excessive waiting time. As s_1 increases, more and more service requests can be served per unit of time, this will bring cloud service providers more and more revenues and profits. Furthermore, the revenues reach the maximum value when $F_{W_1}(D)$ is equal to 1, but the costs will continue to grow

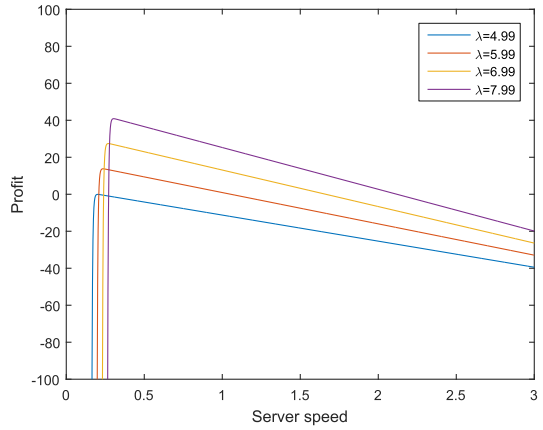


FIGURE 4. Profit G_1 versus s_1 and λ_1.

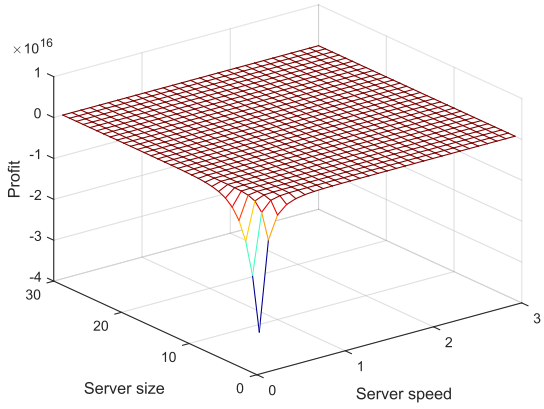


FIGURE 5. Profit G_1 versus s_1 and m_1.

as s_1 further increases, which will result in a profits decline instead. This is because the execution speed of servers exceed the maximum speed required to execute the service requests.

3) OPTIMAL SIZE AND SPEED

According to the previous analyses, it is reasonable to think that both the influences of m_1 and s_1 can result in a much higher increment in the optimal profit than the one discussed in the previous subsections. Therefore, we aim to find the best combination of m_1 and s_1 such that the profit G_1 is maximized.

Figure.5 shows the surface of profit G_1 as a function of m_1 and s_1 with $\lambda_1 = 5.99$. For the reason that the surface is convex, we adopt gradient descent algorithm to find m_1 and s_1 such that the gradient of $G_1(m_1, s_1)$ shown in Eq.(32) is equal to 0, and then the optimal profit is obtained.

$$\nabla G_1(m_1, s_1) = \left\{ \frac{\partial G_1(m_1, s_1)}{\partial m_1}, \frac{\partial G_1(m_1, s_1)}{\partial s_1} \right\} \tag{32}$$

where $\nabla G_1(m_1, s_1)$ is a vector, in which $\partial G_1(m_1, s_1)/\partial m_1$ and $\partial G_1(m_1, s_1)/\partial s_1$ are already derived in the previous subsection. Notice that, since gradient descent algorithm used to solve the minimization problem, while the profit G_1 should be maximized, we multiply $G_1(m_1, s_1)$ by -1 as the objective function of gradient descent algorithm. Moreover, to accelerate the convergence speed of the algorithm, the Arjimo search method is adopted to adjust step automatically [37].

Algorithm 1 Optimal Profit in Multi-Server System S_1

Input: $\lambda_1, a_1, \bar{r}, \alpha, \beta, \delta, \xi, P^*$ and D
Output: the optimal number of servers m_1, optimal execution speed s_1 and optimal profit G_1

begin
  Set the interval of server size $\left[m_1^{min}, m_1^{max}\right]$ and server speed $\left[s_1^{min}, s_1^{max}\right]$;
  count $\leftarrow 1, \theta \leftarrow 10^{-5}$;
  select (m_1^{max}, s_1^{max}) as the start node;
  $m_1^{curr} \leftarrow m_1^{max}, s_1^{curr} \leftarrow s_1^{max}$;
  while *count* < *Max number of iterations* **do**
    $\nabla G_1 \leftarrow$ calculate gradient of G_1;
    $\|-\nabla G_1\| \leftarrow$ calculate 2-norm of $-\nabla G_1$;
    if $\|-\nabla G_1\| < \theta$ **then**
      $m_1^{opt} \leftarrow m_1^{curr}$;
      $s_1^{opt} \leftarrow s_1^{curr}$;
      break;
    else
      $m_1^{curr}, s_1^{curr} \leftarrow$ update the current solution using Armijo search method;
    end
    count $\leftarrow$ count + 1;
  end
  $G_1^{opt} \leftarrow$ calculate optimal profit using m_1^{opt}, s_1^{opt};
  return $m_1^{opt}, s_1^{opt}, G_1^{opt}$
end

The algorithm procedure is given in Algorithm 1. During the calculation, we find the optimal profit is $G_1 = 55.7066$ with $m_1 = 7.559$ and $s_1 = 0.9368$ respectively. By following the same synthesis to the cases with $\lambda_1 = 4.99, 6.99, 7.99$, the optimal profits are $G_1 = 46.286, 63.9184, 75.3548$ with $m_1 = 5.9312, 5.9784, 8.9098$ and $s_1 = 0.9876, 1.3223, 1.0038$ respectively.

B. MAXIMIZE PROFIT IN THE SECOND MULTI-SERVER SYSTEM

By selecting the appropriate m_1 and s_1, the optimal profit G_1 is obtained in the first multi-server system. On this basis, we aim to find the optimal m_2 and s_2 such that the profit G_2 in the second multi-server system is maximized.

1) OPTIMAL SIZE

To obtain the maximum profit in the S_2, the influence of the number of servers m_2 on G_2 is first discussed. Since G_2 is a function of m_1, m_2 and s_1, s_2, and optimal m_1, s_1 are obtained in the previous subsection, we adopt partial derivative to find the optimal m_2, which can be described as follow.

$$\frac{\partial G_2(m_1, m_2, s_1, s_2)}{\partial m_2} = \frac{\partial \varepsilon_2}{\partial m_2} - \frac{\partial C_2}{\partial m_2} = 0 \tag{33}$$

where

$$\frac{\partial \varepsilon_2}{\partial m_2} = \lambda_1 a_2 \bar{r} F_{W_1}(D) \frac{\partial F_W(D)}{\partial m_2}$$

and

$$\frac{\partial C_2}{\partial m_2} = \beta + \delta P^* + \lambda_1 \bar{r} \delta \xi s_2^{\alpha-1} F_{W_1}(D) \frac{\partial F_W(D)}{\partial m_2}$$

Apply the approximation method to $F_W(D)$, which is demonstrated in the first multi-server system, we have

$$F_W(D) \approx A'' \left(1 - e^{-(1-\rho_1)m_1\mu_1^s D}\right) + B'' \left(1 - e^{-(1-\rho_2)m_2\mu_2^s D}\right) \quad (34)$$

where

$$A'' = \frac{1 - P_{q_2}}{\sqrt{2\pi m_1}(1-\rho_1)\left(\frac{e^{\rho_1}}{e\rho_1}\right)^{m_1} + 1}$$

$$B'' = P_{q_2}\left[1 - \frac{1 - \lambda_1 F_{W_1}(D)}{\sqrt{2\pi m_1}(1-\rho_1)\left(\frac{e^{\rho_1}}{e\rho_1}\right)^{m_1} + 1} \times \frac{1}{(1-\rho_1)m_1\mu_1^s - (1-\rho_2)m_2\mu_2^s}\right]$$

For convenience, we set $\gamma_1 = (1-\rho_2)m_2\mu_2^s$, $\gamma_2 = \sqrt{2\pi m_2}(1-\rho_2)\varphi_2$ and $\varphi_2 = \left(\frac{e^{\rho_2}}{e\rho_2}\right)^{m_2}$, then we have

$$\frac{\partial \gamma_1}{\partial m_2} = \mu_2^s$$
$$\frac{\partial \gamma_2}{\partial m_2} = \sqrt{2\pi m_2}\varphi_2\left[\frac{1}{2m_2}(1+\rho_2) - \ln\rho_2(1-\rho_2)\right] \quad (35)$$

Based on Eq.(35), we get

$$\frac{\partial F_W(D)}{\partial m_2} = (1-\sigma_1)\frac{\partial A''}{\partial m_2} + \left(1 - e^{-\gamma_1 D}\right)\frac{\partial B''}{\partial m_2} + \gamma_1\mu_2 D e^{-\gamma_1 D} B'' \quad (36)$$

where

$$\frac{\partial A''}{\partial m_2} = \frac{\frac{\partial \gamma_2}{\partial m_2}}{(\gamma_2+1)^2} \cdot \frac{1}{\sqrt{2\pi m_1}(1-\rho_1)\left(\frac{e^{\rho_1}}{e\rho_1}\right)^{m_1} + 1}$$

$$\frac{\partial B''}{\partial m_2} = \frac{\mu_2^s P_{q_1}\left(1 - \lambda_1 F_{W_1}(D)\right)}{(\gamma_2+1)\left[(1-\rho_1)m_1\mu_1 - \gamma_1\right]} - \frac{\frac{\partial \gamma_2}{\partial m_2}}{(\gamma_2+1)^2}\left[1 - \frac{P_{q_1}\left(1 - \lambda_1 F_{W_1}(D)\right)}{(1-\rho_1)m_1\mu_1^s - \gamma_1}\right]$$

Considering the optimal m_1, s_1 obtained in the first stage, we have the upper limit of $m_2\mu_2^s$ is $m_1\mu_1^s = m_1 s_1/\bar{r}$ when all of incoming service requests to S_1 can be served within the deadline D, i.e. $F_{W_1}(D) = 1$. However, if the waiting time of some service requests exceeds the deadline, the arrival rate of service requests to S_2 will reduce to $\bar{k}F_{W_1}(D)\mu_1^s$, which will cause a decline in the upper limit of $m_2\mu_2$. According to Eq.(47) in Appendix A, the upper limit of $m_2\mu_2^s$ will be restricted to $m_1\mu_1^s - \lambda_1\left(1 - F_{W_1}(D)\right)$. Moreover, to guarantee the ergodicity of multi-server system S_2, $\rho_2 = \bar{k}F_{W_1}(D)\mu_1^s/m_2\mu_2^s$ should be less than 1. Therefore, given a specific s_2, we have

$$\frac{\lambda_1 F_{W_1}(D)\bar{r}}{s_2} \le m_2 \le \frac{\left[m_1\mu_1^s - \lambda_1\left(1 - F_{W_1}(D)\right)\right]\bar{r}}{s_2}$$

Apparently, the analytical solution of Eq.(33) cannot be calculated. Given fixed s_2 and λ_1, the curves of profit G_2 as a function of m_2 are drawn in Figure.6. For $\lambda_1 = 4.99, 5.99, 6.99, 7.99$, we find that $\partial G_2/\partial m_2$ is a decreasing function of m_2. In this case, we can adopt the standard bisection method to obtain the optimal m_2 such that G_2 is maximized.

Therefore, using the analytical method, the optimal value of m_2 is 1.768, 2.132, 2.426, 2.766 and the corresponding profit G_2 is 27.869, 33.8031, 39.2325, 44.9157 respectively.

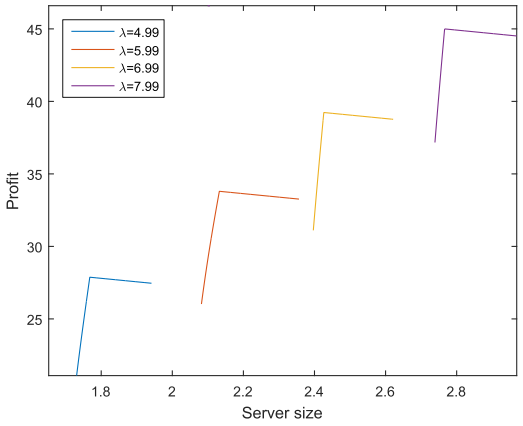


FIGURE 6. Profit G_2 versus m_2 and λ_1.

2) OPTIMAL SPEED

Now consider the influence of execution speed s_2 on G_2, to maximize the profit, we adopt the partial derivative of G_2 with respect to s_2, which is described as follow.

$$\frac{\partial G_2(m_1, m_2, s_1, s_2)}{\partial s_2} = \frac{\partial \varepsilon_2}{\partial s_2} - \frac{\partial C_2}{\partial s_2} = 0 \quad (37)$$

where

$$\frac{\partial \varepsilon_2}{\partial s_2} = \lambda_1 a_2 \bar{r} F_{W_1}(D) \frac{\partial F_W(D)}{\partial s_2}$$

and

$$\frac{\partial C_2}{\partial s_2} = \lambda_1 \bar{r}\delta\xi F_{W_1}(D)\left[(\alpha-1)s_2^{\alpha-2}F_W(D) + s_2^{\alpha-1}\frac{\partial F_W(D)}{\partial s_2}\right]$$

Based on the approximated form of $F_W(D)$ shown in Eq.(34), we have

$$\frac{\partial \gamma_1}{\partial s_2} = \frac{m_2}{\bar{r}}$$
$$\frac{\partial \gamma_2}{\partial s_2} = \frac{\sqrt{2\pi m_2}\varphi_2}{s_2}\left(\rho_2 + m_2(1-\rho_2)^2\right) \quad (38)$$

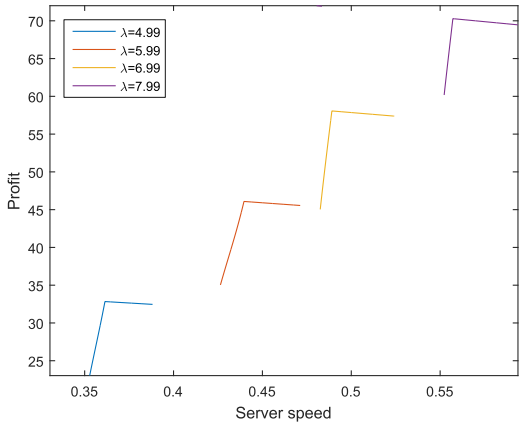


FIGURE 7. Profit G_2 versus s_2 and λ_1.

Then, we have

$$\frac{\partial F_W(D)}{\partial s_2} = (1-\sigma_1)\frac{\partial A''}{\partial s_2} + \left(1-e^{-\gamma_1 D}\right)\frac{\partial B''}{\partial s_2} + \frac{1}{\bar{r}} m_2\gamma_1 D e^{-\gamma_1 D} B'' \quad (39)$$

where

$$\frac{\partial A''}{\partial s_2} = \frac{\frac{\partial \gamma_2}{\partial s_2}}{(\gamma_2+1)^2} \cdot \frac{1}{\sqrt{2\pi m_1}(1-\rho_1)\left(\frac{e\rho_1}{e\rho_1}\right)^{m_1}+1}$$

$$\frac{\partial B''}{\partial s_2} = \frac{m_2\mu_2^s P_{q_1}\left(1-\lambda_1 F_{W_1}(D)\right)}{\bar{r}(\gamma_2+1)\left[(1-\rho_1)m_1\mu_1-\gamma_1\right]} - \frac{\frac{\partial \gamma_2}{\partial s_2}}{(\gamma_2+1)^2}\left[1-\frac{P_{q_1}\left(1-\lambda_1 F_{W_1}(D)\right)}{(1-\rho_1)m_1\mu_1^s-\gamma_1}\right]$$

Apparently, the analytical solution of Eq.(37) cannot be calculated. Given fixed m_2 and λ_1, the curves of profit G_2 as a function of s_2 are drawn in Figure.7. For $\lambda_1 = 4.99, 5.99, 6.99$, we find that $\partial G_2/\partial s_2$ is a decreasing function of m_2. In this case, we can adopt the standard bisection method to obtain the optimal s_2 such that G_2 is maximized.

Therefore, using the analytical method, the optimal value of s_2 is 0.3615, 0.4397, 0.4891, 0.5572 and the corresponding profit G_2 is 32.835, 46.0814, 58.0712, 70.2866 respectively.

3) OPTIMAL SIZE AND SPEED

For cloud service providers, the maximization of the profits are their primary concern. Moreover, they also concern the potential benefits. In other words, when the customers are satisfied with the QoS, they have a higher probability to recommend the cloud computing platform to other customers, then these potential customers will bring more profits to cloud service providers. However, if the customers feel dissatisfied, they are less likely to recommend the cloud computing platform to other customers, then the corresponding profits obtained by cloud service providers will decrease. In this paper, we choose the total waiting time of customers in multi-stage multi-server queue systems to measure the satisfaction of customers. When the waiting time exceeds the deadline D, the customers feel dissatisfied, and vice versa. Therefore, we aim to maximize the total profits on the premise of increasing the number of customers who are served within the deadline as much as possible, the corresponding mathematical programming model is formulated as follow.

$$\begin{aligned} \min\ & f_1(X) = -G_2 = -\varepsilon_2 + C_2 \\ & f_2(X) = 1 - F_W(D) \\ s.t.\ & \begin{cases} m_2\mu_2^s < m_1\mu_1^s - \lambda_1\left(1-F_{W_1}(D)\right) \\ m_2\mu_2^s > \lambda_1 F_{W_1}(D) \\ 0 \le F_W(D) \le 1 \end{cases} \end{aligned} \quad (40)$$

where $X = [m_2, s_2]$. Considering Eq.(40) has two objectives, we adopt NSGA-II algorithm to solve this problem [38], [39], the pseudocode of the algorithm is described in Algorithm 2.

Algorithm 2 Optimal Profit in Multi-Server System S_2

Input: interval of server size $\left[m_2^{min}, m_2^{max}\right]$, server speed $\left[s_2^{min}, s_2^{max}\right]$
Output: optimal profit G_2 and $F_W(D)$

```
begin
    t ← 1, i ← 1;
    initialize the population in the given interval;
    T_1 ← sort initial population using non-dominated sorting strategy;
    while t < Max number of iterations do
        P_t ← create parent by T_t using tournament selection;
        Q_t ← create offspring by P_t using selection, crossover and mutation;
        R_t ← P_t ∪ Q_t;
        F_t ← calculate all non-dominated fronts of R_t;
        P_{t+1} ← ∅;
        while |P_{t+1}| ∪ |F_t^i| ≤ N do
            F_t^i ← select ith non-dominated front in F_t using crowding distance sorting strategy;
            P_{t+1} ← P_{t+1} ∪ F_t^i;
            i ← i + 1;
        end
        T_{t+1} ← P_{t+1} ∪ F_t [1 : (N − |P_{t+1}|)];
        t ← t + 1;
    end
end
```

In Algorithm 2, N is the population size, and $|\cdot|$ represents the number of individuals in the corresponding set. By adopting the NSGA-II algorithm, we obtain the non-dominated solution as the optimal frontier of the optimization problem, which is shown in Figure.8.

As can be seen from Figure.8, the optimization of $f_1(X)$ and $f_2(X)$ can not be achieved simultaneously. On the basis, we randomly select a pareto optimal solution in the optimal frontier, i.e. $f_1(X) = -57.1, f_2(X) = 0.00197$, then the

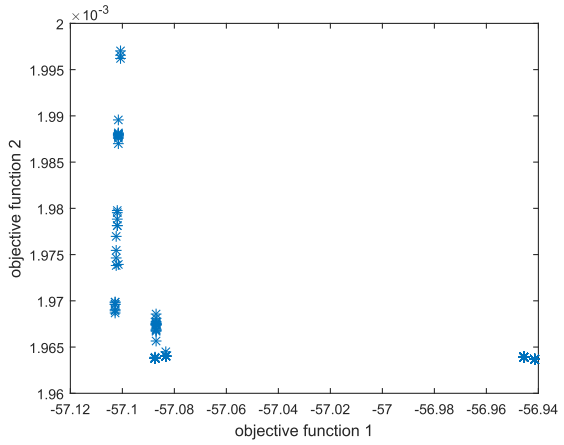


FIGURE 8. Pareto optimal solutions of G_2 and $F_W(D)$.

maximum profit in multiple multi-server systems is 57.1 with 99.8% service requests being served within the deadline.

C. GENERAL CASE WITH n-STAGE MULTI-SERVER QUEUE SYSTEMS

The profit maximization scheme in cloud computing platform with two-stage multi-server queue systems has been fully discussed. In this section, such method is further extended to the general case with n-stage multi-server queue systems. According to Eq.(18), we find that the profit in each multi-server system depends only on the parameters of the current system and the systems before that, not on the systems after that. It is obvious, because no matter what stage the customers are waiting for, once their total waiting time exceeds the deadline, they will depart from the multistage multi-server queue systems even if their tasks have not been fully executed. Therefore, there are no need to pay for the service requests (sub-tasks) which are not served, then the profits obtained in the corresponding multi-server systems are no need to analyze as well. On this basis, we summarize the execution procedure of the algorithm in Algorithm 3.

Algorithm 3 Optimal Profit in n Multi-Server Systems

Input: $\lambda_1, a_1, \bar{r}, \alpha, \beta, \delta, \xi, P^*$ and D
Output: optimal number of servers $m_1, m_2, \cdots, m_n$, optimal execution speed $s_1, s_2, \cdots, s_n$ and optimal profit $G_1, G_2, \cdots, G_n$

begin
  $m_1^{opt}, s_1^{opt}, G_1^{opt} \leftarrow$ call Algorithm 1;
  $S^{curr} \leftarrow S_1$;
  $G^{opt} \leftarrow G_1^{opt}$;
  i $\leftarrow 2$;
  while $i < n$ **do**
    $S^{curr} \leftarrow S^{curr} \cup S_i$;
    $f_W^{curr}(t) \leftarrow$ calculate pdf of total waiting time in S^{curr} using Theorem 1;
    $m_i^{opt}, s_i^{opt}, G_i^{opt} \leftarrow$ call Algorithm 2;
    $G^{opt} \leftarrow G^{opt} + G_i^{opt}$;
    i $\leftarrow$ i $+ 1$;
  end
end

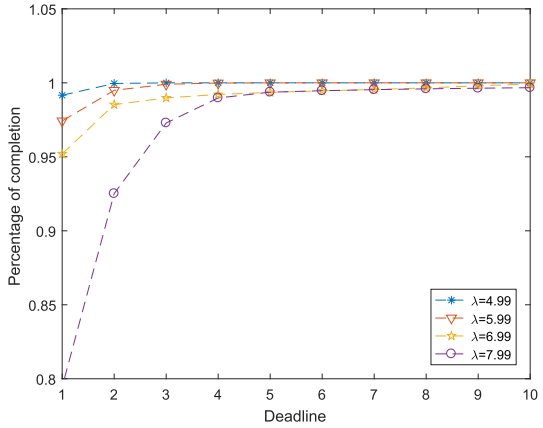


FIGURE 9. Percentage of executed service requests under deadline constraint.

V. PERFORMANCE ANALYSIS

Based on the analysis in the previous section, we find that the percentage of service requests which are served within the deadline can not only be affected by m_1, m_2 and s_1, s_2, but also be affected by the arrival rate of service requests λ_1 and deadline D. In our first group of simulations, we aim to demonstrate the variations of the percentage of service requests which are served within the deadline and the total profit with an increasing level of deadline under different arrival rates, the corresponding results are shown in Figure.9 and 10. For the multistage multi-server queue systems with $\lambda_1 = 4.99, 5.99, 6.99, 7.99$ respectively, the percentage of service requests which are served within the deadline and the total profit both increased with the deadline. The reason lies in the fact that more service requests can be served as the deadline increases, which will bring more revenues to cloud service providers. Moreover, with the decrease in λ_1, the total profit are decreased, while the percentage of service requests which are served within the deadline are increased for a fixed deadline. This is because when the arrival rate of service requests is low, the servers only suffer few pressure in the multistage multi-server queue systems, then the newly arrived service requests have a higher probability to be served immediately, which can effectively reduce the waiting time to satisfy the deadline constraint.

In our second group of simulations, we aim to demonstrate the variations of the percentage of service requests which are served within the deadline and the total profit with the arrival rate of service requests increasing under different deadlines, the corresponding results are shown in Figure.11 and 12. For the multistage multi-server queue systems with $D = 1, 4, 7, 10$ respectively, the total profit increase with λ_1, while the percentage of service requests which are served

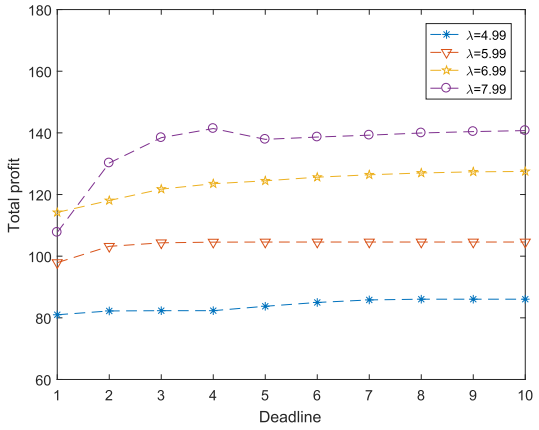


FIGURE 10. Optimal profit versus the deadline.

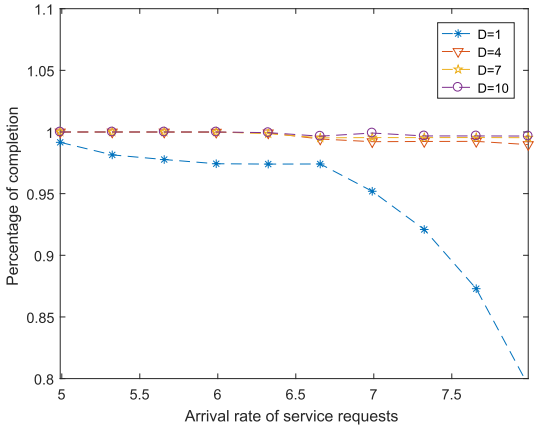


FIGURE 11. Percentage of executed service requests versus λ_1.

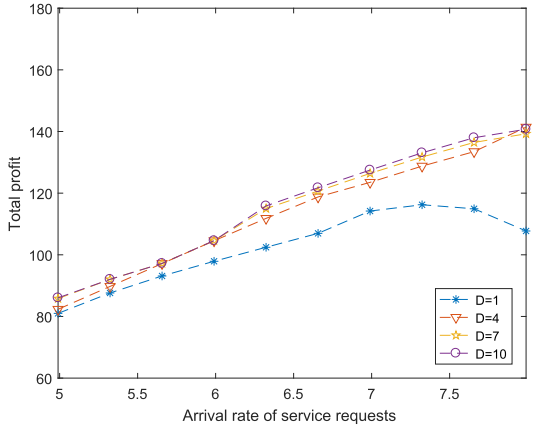


FIGURE 12. Optimal profit versus λ_1.

within the deadline change in the opposite direction. Notice that, for $D = 4, 7, 10$, the variations of total profits and the percentage of service requests which are served within the deadline are close to each other, and for $D = 1$, such variation has a significant difference. This is because the deadline is long enough in the former case to allow almost all of the service requests to be served, while such deadline is too short in the latter case to allow enough service requests to be served. Therefore, the total profit and the percentage of service requests which are served within the deadline in the latter case both are much less than the one obtained in the former case. Moreover, when $D = 1$, as λ_1 further increases, the total profit decreases instead, for the reason that the increase in revenue is not enough to compensate for the increase in cost. Hence, a trade-off should be addressed between the maximization of profit and percentage of service requests which are served within the deadline in this case.

VI. CONCLUSION

In this paper, we aim to achieve the profit maximization for cloud service providers on the premise of fulfilling the task execution requirements of customers under deadline constraint. Considering that each task can be separated into numbers of sub-tasks (service requests) with a successive execution relationship, we present a cloud computing platform consists of multistage multi-server queue systems, each system devotes to serve unique type of service requests in a stage. On this basis, there arises a contradiction between maximizing the profit of cloud service provider and minimizing the losses of customers due to excessive waiting time. To solve the problem, we propose a heuristic algorithm for the reason that the analytical solution is very difficult to obtain. At last, the performance evaluation of the proposed method has been investigated by a series of numerical simulations, the results show the dynamic characteristics of the proposed profit maximization scheme with an increasing level of deadline and arrival rate of service requests respectively.

APPENDIX A
PROOF OF THEOREM III.1

Proof: Given $f_{W_1}(t)$ and $f_{W_2}(t)$ as the pdf of waiting time of a service request spent in the first and second multi-server system respectively, the total waiting time $W = W_1 + W_2$ is a random variable whose pdf can be calculated by the convolution of $f_{W_1}(t)$ and $f_{W_2}(t)$, namely

$$\begin{aligned} f_W(t) &= f_{W_1+W_2}(t) \\ &= f_{W_1}(t) \otimes f_{W_2}(t) \\ &= \int_0^\infty f_{W_1}(\tau) f_{W_2}(t-\tau)\, d\tau. \end{aligned} \tag{41}$$

where $\otimes$ represent the convolution operation. Substituting $f_{W_1}(t)$ and $f_{W_2}(t)$ into Eq.(41), we have

$$\begin{aligned} &\int_0^\infty f_{W_1}(\tau) f_{W_2}(t-\tau)\, d\tau \\ &= \int_0^\infty \Bigg[\prod_{i=1}^{2} \left(1 - P_{q_i}\right) u(\tau)\, u(t-\tau) \\ &\quad + \left(1 - P_{q_1}\right) u(\tau) \cdot m_2 \mu_2^s p_{m_2} e^{-(1-\rho_2) m_2 \mu_2^s (t-\tau)} \\ &\quad + \left(1 - P_{q_2}\right) u(t-\tau) \cdot m_1 \mu_1^s p_{m_1} e^{-(1-\rho_1) m_1 \mu_1^s \tau} \\ &\quad + \prod_{i=1}^{2} \left(m_i \mu_i^s p_{m_i}\right) e^{-\left[(1-\rho_1) m_1 \mu_1^s \tau + (1-\rho_2) m_2 \mu_2^s (t-\tau)\right]} \Bigg] d\tau \end{aligned} \tag{42}$$

As can be seen from Eq.(42), $f_W(t)$ is separated into four parts. For the first part, since $u(\tau)$ and $u(t-\tau)$ both are unit impulse function, and they set nonzero values at different time. Hence, we have $u(\tau)\,u(t-\tau) = 0$, which also means

$$\int_0^\infty \prod_{i=1}^{2} \left(1-P_{q_i}\right) u(\tau)\,u(t-\tau)d\tau = 0 \tag{43}$$

For the second and third part, we have

$$\begin{aligned}&\int_0^\infty \left(1-P_{q_1}\right) u(\tau)\cdot m_2\mu_2^s p_{m_2} e^{-(1-\rho_2)m_2\mu_2^s(t-\tau)}d\tau \\ &\quad = \left(1-P_{q_1}\right) m_2\mu_2^s p_{m_2} e^{-(1-\rho_2)m_2\mu_2^s t} \end{aligned}\tag{44}$$

$$\begin{aligned}&\int_0^\infty \left(1-P_{q_2}\right) u(t-\tau)\cdot m_1\mu_1^s p_{m_1} e^{-(1-\rho_1)m_1\mu_1^s\tau}d\tau \\ &\quad = \left(1-P_{q_2}\right) m_1\mu_1^s p_{m_1} e^{-(1-\rho_1)m_1\mu_1^s t} \end{aligned}\tag{45}$$

And for the last part, we have

$$\begin{aligned}&\int_0^\infty \prod_{i=1}^{2} \left(m_i\mu_i^s p_{m_i}\right) e^{-\left[(1-\rho_1)m_1\mu_1^s\tau+(1-\rho_2)m_2\mu_2^s(t-\tau)\right]}d\tau \\ &= \prod_{i=1}^{2} \left(m_i\mu_i^s p_{m_i}\right) e^{-(1-\rho_2)m_2\mu_2^s t} \\ &\quad \times \int_0^\infty e^{-\left[(1-\rho_1)m_1\mu_1^s-(1-\rho_2)m_2\mu_2^s\right]\tau}d\tau \\ &= \frac{\prod_{i=1}^{2}\left(m_i\mu_i^s p_{m_i}\right)e^{-(1-\rho_2)m_2\mu_2^s t}}{(1-\rho_2)m_2\mu_2^s-(1-\rho_1)m_1\mu_1^s}\; e^{-\left[(1-\rho_1)m_1\mu_1^s-(1-\rho_2)m_2\mu_2^s\right]\tau}\Big|_0^\infty \end{aligned}\tag{46}$$

Obviously, to guarantee the boundedness of Eq.(46), $(1-\rho_1)m_1\mu_1^s - (1-\rho_2)m_2\mu_2^s$ must be greater than 0. Hence, we have

$$\begin{aligned}&(1-\rho_1)\,m_1\mu_1^s - (1-\rho_2)\,m_2\mu_2^s \\ &\quad = m_1\mu_1^s - \lambda_1 - m_2\mu_2^s + \bar{k}\mu_1^s \\ &\quad = m_1\mu_1^s - m_2\mu_2^s \\ &\quad > 0 \end{aligned}\tag{47}$$

Namely, if and only if the condition $m_2\mu_2^s < m_1\mu_1^s$ is satisfied, the following equation can be obtained.

$$\begin{aligned}&\int_0^\infty \prod_{i=1}^{2} \left(m_i\mu_i^s p_{m_i}\right) e^{-\left[(1-\rho_1)m_1\mu_1^s\tau+(1-\rho_2)m_2\mu_2^s(t-\tau)\right]}d\tau \\ &\qquad = \frac{\prod_{i=1}^{2}\left(m_i\mu_i^s p_{m_i}\right)}{(1-\rho_1)m_1\mu_1^s-(1-\rho_2)m_2\mu_2^s} e^{-(1-\rho_2)m_2\mu_2^s t} \end{aligned}\tag{48}$$

Substituting Eq.(43), (44), (45) and (48) into Eq.(42), the pdf of total waiting time $f_W(t)$ can be obtained, which proves the theorem.

□

APPENDIX B
PROOF OF THEOREM III.2

Proof: The pdf of the waiting time W_1, W_2 and W are

$$\begin{aligned} f_{W_i}(t) &= \left(1-P_{q_i}\right) u(t) + m_i\mu_i p_{m_i} e^{-(1-\rho_i)m_i\mu_i^s t} \\ f_W(t) &= Ae^{-(1-\rho_1)m_1\mu_1^s t} + Be^{-(1-\rho_2)m_2\mu_2^s t} \end{aligned}$$

where $i \in \{1,2\}$. Since W_1, W_2 and W are random variables, then $R_1(r, W_1)$ and $R_2(r, W_1, W_2)$ can be considered as random variables as well. Therefore, we can calculate the expectations of $R_1(r, W_1)$ and $R_2(r, W_1, W_2)$ by the following equations.

$$\begin{aligned} R_1(r) &= E\left(R_1(r, W_1)\right) \\ &= \int_0^\infty f_{W_1}(t)\, a_1 r dt \\ &= a_1 r \int_0^\infty \left[\left(1-P_{q_1}\right) u(t) + m_1\mu_1^s p_{m_1} e^{-(1-\rho_1)m_1\mu_1^s t}\right] dt \\ &= a_1 r \int_0^D \left[\left(1-P_{q_1}\right) u(t) + m_1\mu_1^s p_{m_1} e^{-(1-\rho_1)m_1\mu_1^s t}\right] dt \\ &= a_1 r \left[\left(1-P_{q_1}\right) - \frac{p_{m_1}}{1-\rho_1}\left(e^{-(1-\rho_1)m_1\mu_1^s D} - 1\right)\right] \\ &= a_1 r \left(1 - \frac{p_{m_1}}{1-\rho_1} e^{-(1-\rho_1)m_1\mu_1^s D}\right) \end{aligned}$$

and

$$\begin{aligned} R_2(r) &= E\left(R_2(r, W_1, W_2)\right) \\ &= \int_0^\infty f_{W_2}(t)\, a_2 r dt \\ &= a_2 r \int_0^{D-W_1} f_{W_2}(t)\, dt \\ &= a_2 r P\left(0 \le W_2 \le D - W_1\right) \\ &= a_2 r \left[P\left(W_2 \le D - W_1\right) - P\left(W_2 \le 0\right)\right] \end{aligned}$$

where $P(\cdot)$ is the cumulative distribution function (cdf). Since W_2 is the waiting time of a service request in multi-server system S_2, it is greater than 0. Therefore, we get $P(W_2 \le 0) = 0$. Furthermore, since $P(W_2 \le D - W_1)$ is equivalent to $P(W_1 + W_2 \le D)$, then we have

$$\begin{aligned} R_2(r) &= a_2 r P\left(W_1 + W_2 \le D\right) \\ &= a_2 r \int_0^D f_W(t)\, dt \\ &= a_2 r \int_0^D \left[Ae^{-(1-\rho_1)m_1\mu_1^s t} + Be^{-(1-\rho_2)m_2\mu_2^s t}\right] dt \\ &= a_2 r \left[\frac{A}{(1-\rho_1)\,m_1\mu_1^s}\left(1 - e^{-(1-\rho_1)m_1\mu_1^s D}\right)\right. \\ &\quad \left. + \frac{B}{(1-\rho_2)\,m_2\mu_2^s}\left(1 - e^{-(1-\rho_2)m_2\mu_2^s D}\right)\right] \\ &= a_2 r \left(A'\left(1 - e^{-(1-\rho_1)m_1\mu_1^s D}\right)\right. \\ &\quad \left. + B'\left(1 - e^{-(1-\rho_2)m_2\mu_2^s D}\right)\right) \end{aligned}$$

Notice that, the task execution requirement r follows the exponential distribution. On this basis, the expected charge

of a service request in multi-server system S_1 and S_2 can be calculated as follow.

$$\begin{aligned}\bar{R}_1 &= E(R_1(r)) \\ &= \int_0^\infty \frac{1}{\bar{r}} e^{-z/\bar{r}} R_1(z) dz \\ &= \frac{a_1}{\bar{r}} \left(1 - \frac{p_{m_1}}{1-\rho_1} e^{-(1-\rho_1)m_1\mu_1^s D}\right) \int_0^\infty e^{-z/\bar{r}} z dz \\ &= a_1 \bar{r} \left(1 - \frac{p_{m_1}}{1-\rho_1} e^{-(1-\rho_1)m_1\mu_1^s D}\right)\end{aligned}$$

and

$$\begin{aligned}\bar{R}_2 &= E(R_2(r)) \\ &= \int_0^\infty \frac{1}{\bar{r}} e^{-z/\bar{r}} R_2(z) dz \\ &= \frac{a_2}{\bar{r}} \left(A' \left(1 - e^{-(1-\rho_1)m_1\mu_1^s D}\right)\right. \\ &\quad \left. + B' \left(1 - e^{-(1-\rho_2)m_2\mu_2^s D}\right)\right) \int_0^\infty e^{-z/\bar{r}} z dz \\ &= a_2 \bar{r} \left(A' \left(1 - e^{-(1-\rho_1)m_1\mu_1^s D}\right)\right. \\ &\quad \left. + B' \left(1 - e^{-(1-\rho_2)m_2\mu_2^s D}\right)\right)\end{aligned}$$

This proves the theorem. □

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] K. Hwang, J. Dongarra, and G. Fox, *Cloud Computing and Distributed Systems 2e: From Parallel Processing to the Internet of Things*, 2nd ed. Amsterdam, The Netherlands: Morgan Kaufmann, 2018. [Online]. Available: https://bib-pubdb1.desy.de/record/407873

[2] M. Armbrust and A. Fox, "Above the clouds: A berkeley view of cloud computing," Dept. EECS, Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, Feb. 2009. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[3] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST Special Publication, Gaithersburg, MD, USA, Tech. Rep., Jan. 2011, vol. 800, p. 145.

[4] D. Durkee, "Why cloud computing will never be free," *ACM Queue*, vol. 53, no. 4, pp. 62–69, 2010.

[5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[6] S. Nesmachnow, S. Iturriaga, and B. Dorronsoro, "Efficient heuristics for profit optimization of virtual cloud brokers," *IEEE Comput. Intell. Mag.*, vol. 10, no. 1, pp. 33–43, Feb. 2015.

[7] L. Wu, S. K. Garg, and R. Buyya, "SLA-based admission control for a software-as-a-service provider in cloud computing environments," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1280–1299, Sep. 2012.

[8] 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, Jun. 2013.

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

[10] P. Cong, L. Li, J. Zhou, K. Cao, T.Wei, M. Chen, and S. Hu, "Profit-driven dynamic cloud pricing for multiserver systems considering user perceived value," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 12, pp. 2742–2756, 2018.

[11] J. Mei, K. Li, and K. Li, "Customer-satisfaction-aware optimal multiserver configuration for profit maximization in cloud computing," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 1, pp. 17–29, Jan. 2017.

[12] K. Li, J. Mei, and K. Li, "A fund-constrained investment scheme for profit maximization in cloud computing," *IEEE Trans. Services Comput.*, vol. 11, no. 6, pp. 893–907, Nov. 2018.

[13] N. K. Boots and H. Tijms, "An $M/M/c$ queue with impatient customers," *TOP, Off. J. Spanish Soc. Statist. Oper. Res.*, vol. 7, no. 2, pp. 213–220, 1999.

[14] A. Verma and S. Kaushal, "Deadline constraint heuristic-based genetic algorithm for workflow scheduling in cloud," *Int. J. Grid Utility Comput.*, vol. 5, no. 2, pp. 96–106, 2014.

[15] Y.-S. Chang, C.-T. Fan, R.-K. Sheu, S.-R. Jhu, and S.-M. Yuan, "An agent-based workflow scheduling mechanism with deadline constraint on hybrid cloud environment," *Int. J. Commun. Syst.*, vol. 31, no. 1, p. e3401, Jan. 2018.

[16] A. Deldari, M. Naghibzadeh, and S. Abrishami, "CCA: A deadline-constrained workflow scheduling algorithm for multicore resources on the cloud," *J. Supercomput.*, vol. 73, no. 2, pp. 756–781, Feb. 2017.

[17] X. Zou, L. Zhang, and Q. Zhang, "A biobjective optimization model for deadline satisfaction in line-of-balance scheduling with work interruptions consideration," *Math. Problems Eng.*, vol. 2018, no. 5, pp. 6534021.1–6534021.12, 2018.

[18] B. Li, Y. Pei, H. Wu, and B. Shen, "Resource availability-aware advance reservation for parallel jobs with deadlines," *J. Supercomput.*, vol. 68, no. 2, pp. 798–819, May 2014.

[19] X. Li, L. Qian, and R. Ruiz, "Cloud workflow scheduling with deadlines and time slot availability," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 329–340, Mar. 2018.

[20] L.-C. Canon, A. K. W. Chang, Y. Robert, and F. Vivien, "Scheduling independent stochastic tasks under deadline and budget constraints," *Int. J. High Perform. Comput. Appl.*, vol. 34, no. 2, pp. 246–264, 2019.

[21] C.-H. Lan, "The design of multiple production lines under deadline constraint," *Int. J. Prod. Econ.*, vol. 106, no. 1, pp. 191–203, Mar. 2007.

[22] L.-H. Su, "Scheduling on identical parallel machines to minimize total completion time with deadline and machine eligibility constraints," *Int. J. Adv. Manuf. Technol.*, vol. 40, nos. 5–6, pp. 572–581, Jan. 2009.

[23] P. Toktaş-Palut and F. Ülengin, "Modeling a supply chain as a queuing system," *IFAC Proc. Volumes*, vol. 43, no. 17, pp. 242–247, 2010.

[24] V. Thangaraj and S. Vanitha, "M/G/1 queue with two-stage heterogeneous service compulsory server vacation and random breakdowns," *Int. J. Contemp. Math. Sci.*, vol. 5, pp. 307–322, Jan. 2010.

[25] S. Ramasamy, O. A. Daman, and S. Sani, "Discrete-time Geo/G/2 queue under a serial and parallel queue disciplines," *IAENG Int. J. Appl. Math.*, vol. 45, no. 4, pp. 354–363, 2015.

[26] S. M. Sundari and S. Srinivasan, "Analysis of $M/G/1$ feedback queue with three stage and multiple server vacation," *Appl. Math. Sci.*, vol. 6, nos. 125–128, pp. 6221–6240, 2012.

[27] A. S. Ajiboye and K. A. Saminu, "A multi-stage queue approach to solving customer congestion problem in a restaurant," *Open J. Statist.*, vol. 8, no. 2, pp. 302–316, 2018.

[28] C.-F. Li, "Cloud computing system management under flat rate pricing," *J. Netw. Syst. Manage.*, vol. 19, no. 3, pp. 305–318, Sep. 2011.

[29] R. Bala and S. Carr, "Usage-based pricing of software services under competition," *J. Revenue Pricing Manage.*, vol. 9, no. 3, pp. 204–216, May 2010.

[30] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. C. M. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2014, pp. 118–126.

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

[32] 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 Microsyst.*, vol. 37, no. 1, pp. 99–112, Feb. 2013.

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

[34] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proc. 41st Annu. Conf. Design Automat. (DAC)*, 2004, pp. 868–873.

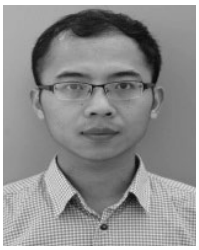
[35] J. Mei, K. Li, A. Ouyang, and K. Li, "A profit maximization scheme with guaranteed quality of service in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3064–3078, Nov. 2015.

[36] K. Li, "Optimal configuration of a multicore server processor for managing the power and performance tradeoff," *J. Supercomput.*, vol. 61, no. 1, pp. 189–214, Jul. 2012.

[37] J. Wang, B. Zhang, Z. Sun, W. Hao, and Q. Sun, "A novel conjugate gradient method with generalized armijo search for efficient training of feedforward neural networks," *Neurocomputing*, vol. 275, pp. 308–316, Jan. 2018.
[38] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.
[39] C. Zhang, "Improved NSGA-II for the multi-objective flexible job-shop scheduling problem," *J. Mech. Eng.*, vol. 46, no. 11, p. 156, 2010.

SIYI CHEN was born in Xiangtan, China, in 1986. He received the B.S. degree from Xiangtan University, China, and the M.S. and Ph.D. degrees in control theory and control engineering from the South China University of Technology, China, in 2016. Since 2017, he has been a Lecturer with the School of Automation and Electronic Information, Xiangtan University. His research interests include service scheduling, cloud computing, and iterative heuristic algorithm.

SINING HUANG was born in Changsha, China, in 1998. He received the B.S. degree in automation from Xiangtan University, Xiangtan, China, in 2016, where he is currently pursuing the M.S. degree with the School of Automation and Electronic Information, Institute of Control Science and Engineering. His current research interest lies in the improvement of intelligent algorithms and the application of intelligent algorithms in service computing.



QIANG LUO was born in Shangrao, China, in 1986. He received the M.S. and Ph.D. degrees in traffic information engineering and control from the School of Civil and Transportation, South China University of Technology, Guangzhou, China, in 2011 and 2014, respectively. His research interests include traffic data analysis, complex traffic network, intelligent transportation systems, and traffic flow assignment model.

JIALING ZHOU was born in Yiyang, China, in 1998. She received the B.S. degree in automation from Xiangtan University, Xiangtan, China, in 2019, where she is currently pursuing the M.S. degree with the School of Automation and Electronic Information, Institute of Control Science and Engineering. Her research interests include intelligent optimization and scheduling algorithms.