Delay and Price Differentiation in Cloud Computing: A Service Model, Supporting Architectures, and Performance

XIAOHU WU, Nanyang Technological University, Singapore & Beijing University of Posts and Telecommunications, China
FRANCESCO DE PELLEGRINI, University of Avignon, France
GIULIANO CASALE, Imperial College London, United Kingdom

Many cloud service providers (CSPs) offer an on-demand service with a small delay. Motivated by the reality of cloud ecosystems, we study non-interruptible services and consider a differentiated service model to complement the existing market by offering multiple service level agreements (SLAs) to satisfy users with different delay tolerance. The model itself is incentive compatible by construction. Two typical architectures are considered to fulfill SLAs: (i) non-preemptive priority queues and (ii) multiple independent groups of servers. We leverage queueing theory to establish guidelines for the resultant market: (a) Under the first architecture, the service model can only improve the revenue marginally over the pure on-demand service model and (b) under the second architecture, we give a closed-form expression of the revenue improvement when a CSP offers two SLAs and derive a condition under which the market is viable. Additionally, under the second architecture, we give an exhaustive search procedure to find the optimal SLA delays and prices when a CSP generally offers multiple SLAs. Numerical results show that the achieved revenue improvement can be significant even if two SLAs are offered. Our results can help CSPs design optimal delay-differentiated services and choose appropriate serving architectures.

CCS Concepts: • Networks → Cloud computing; network economics; • Computing methodologies → Model development and analysis; • Mathematics of computing → Queueing theory;

Additional Key Words and Phrases: Service differentiation, incentive compatible, cloud computing

ACM Reference format:
Xiaohu Wu, Francesco De Pellegrini, and Giuliano Casale. 2023. Delay and Price Differentiation in Cloud Computing: A Service Model, Supporting Architectures, and Performance. ACM Trans. Model. Perform. Eval. Comput. Syst. 8, 3, Article 6 (June 2023), 40 pages.
https://doi.org/10.1145/3592852

1 INTRODUCTION

The Infrastructure-as-a-Service market grew to total $90.9 billion in 2021 [1] and has attracted plenty of users with different purposes to run their applications on cloud servers [2]. As an essential
option, many cloud service providers (CSPs) offer the standard on-demand service with a negligible delay so that users can always access computing resource fast. While delay is a key constraint to resource efficiency [3], users often differ in the sensitivity to it [4, 5]. Price differentiation by delay is an important direction to both satisfy user preference and improve CSP revenue. An example is Amazon Elastic Cloud Compute (EC2), which offers delay-tolerant spot service together with the standard on-demand service, which can closely satisfy the need of users with different delay tolerance; other such CSPs include Microsoft Azure and Google Cloud, whose delay-tolerant services are called low-priority VMs and preemptive VM instances, respectively [11–13, 15]. However, CSPs typically do not reveal operational details behind spot and on-demand services to the public, since this is commercially sensitive information. As an ecosystem, queueing theory is often used to analyze its performance while incentive compatibility is needed to ensure user truthfulness, thus eliminating the unpredictable effect of non-truthful strategic behavior on performance [7, 8].

Abhishek et al. propose a potential way of operating spot and on-demand services and describe the on-demand market by a multi-server queue and the spot market by a preemptive priority queue [9]. Each user may be characterized by an initial willingness-to-pay (WTP) that decreases linearly with the delay. The associated slope c defines how sensitive users are to delay and is called delay–cost type. Users choose to join one market or neither to maximize their surpluses. For the spot market, users bid to access shared servers. Higher bidders can immediately interrupt the service being delivered to lower bidders. Each type of user has an individual service class whose delay relies on the load of higher bidders. Abhishek et al. also give a pricing rule to guarantee the Bayesian–Nash incentive compatibility, i.e., a user will truthfully bid c if others do so [9]. Dierks and Seuken significantly extend the framework of Abhishek et al. by considering additional realistic constraints such as the preemption costs in spot market and the capacity finiteness of on-demand market and derive an easy-to-check condition under which a complementary spot market is viable [10, 11].

Differently from References [9, 10], we consider the following dimensions rooted in the reality and participant’s need of cloud ecosystems. First, the standard on-demand service has existed for years and attracted plenty of users. Its price, denoted by p, is predefined, moderate, and well accepted by these users, although they have potential diversity in delay tolerance. Our characterization of WTP is intended to embody this reality directly, and it defines how much discount is needed to let delay-tolerant users accept delayed services, since we want to complement the existing on-demand service with delayed services and study how this affects the user behavior and CSP gain. Thus, we let the initial WTPs of all users be p, implying their acceptance of on-demand service, and their WTPs still decrease with the delay. Second, the number of user delay–cost types is large, and it may be operationally costly to maintain an individual service level agreement (SLA) for each type of users [17]. Third, we focus on non-preemptive scheduling, i.e., the service is continuously provisioned to every user with no interruption. Admittedly, the interruptible service in References [9, 10] has an advantage that servers are always allocated to the highest bidders to generate a high revenue. However, preemptions often involve saving and recovering the states of preempted servers and can be costly [18, 19]. Users are also burdened with extra complexity while facing interruptible service [20–22]. In this article, we are interested in studying whether a non-interruptible service system is still viable.

The standard on-demand service is designed with the principle of “one size fits all” to satisfy all types of users. To complement this, we propose a model of service offerings based on a limited number of SLAs to provide multiple classes of services. These SLAs include both on-demand service for latency-critical jobs and delayed services for delay-tolerant jobs at lower prices. The fulfillment of SLAs relies on an underlying architecture of servers to process the incoming jobs. Two typical architectures are considered. One architecture is similar to the spot market in
References [9, 10] but is described by a non-preemptive priority queue, called the priority-based sharing (PBS) architecture. The other simply divides servers into multiple groups that independently take charge of different SLAs, called the separated multi-SLAs (SMS) architecture. The service model supported by the PBS architecture (respectively, by the SMS architecture) is called a PBS-based service system (respectively, a SMS-based service system).

The proposed service system may benefit all market participants. Potential users get opportunity to satisfactorily trade their delay tolerance for cheaper service, without extra burden to manage interruptions, since non-preemptive services are offered. The CSP can thus attract more such users from its competitors and establish its reputation. In queueing systems, the larger the delay, the higher the resource utilization. Delay-differentiated services may allow processing more workload than a pure on-demand service, possibly improving its revenue by proper pricing.

In this article, we study the above non-interruptible service system that offers $L$ SLAs. It is dominant-strategy incentive compatible (DSIC) by construction: Every user truthfully reports its delay–cost type, regardless of what others do. Its viability is measured by the ratio of its revenue to the revenue of a pure on-demand service model. The main results of this article are as follows:

(i) Regarding the PBS-based service system, we give an upper bound of the revenue improvement, which is independent of the delay tolerances of users and the job arrival rate of users (see Section 5.4). This bound somehow shows that it can only achieve a marginal revenue improvement. This implies its unviability and the necessity of studying the SMS-based service system.

(ii) Regarding the SMS-based service system that offers two SLAs (i.e., $L = 2$), we derive under mild assumptions the optimal SLA delays and prices to maximize the CSP’s revenue and give a closed-form expression of the revenue improvement. This leads to a condition under which the SMS-based system is viable. The condition shows that the system is viable if the delay-tolerance level of the user population is greater than a threshold that relates to the delay of on-demand service, the average job arrival rate per server, and the second moment of the job service time distribution (see Section 5.5); here numerical results are accompanied to show that the CSP can achieve a significant revenue improvement under a wide range of conditions.

(iii) Regarding the SMS-based service system that offers multiple SLAs (i.e., $L \geq 2$), we give an exhaustive search procedure that finds the optimal SLA prices and delays (see Section 5.6). This allows us to give numerical results to show the performance of the SMS-based system in more cases (e.g., $L \geq 3$), highlighting that (a) the revenue improvement is non-decreasing in $L$ and (b) the improvement in the case of $L = 2$ is of the same order of magnitude as the improvements in the case of $L \geq 3$ (see Section 6).

We note that all these results hold when the WTP functions are linear [9–13, 15]. For analytical tractability, such a case is often studied to get rich insight. Except the second result above, our results can also hold when the WTP functions are concave [15] (Section 3.1.1). Even so, we can see in our numerical experiments that the conclusions for linear WTPs can serve as a guide to the performance in the case with concave WTPs.

The rest of this article is organized as follows. In Section 2, we introduce the related work. We describe the proposed delay-differentiated service model in Section 3. Next, we study in Section 4 the related pricing problems. We describe two architectures in Section 5 to support the service model differently and analyze their performance and optimal parameter configuration; here we also give numerical results for the SMS-based service system with $L = 2$. In Section 6, additional numerical results are given to show the performance of the SMS-based system in more cases. Finally, we conclude this article in Section 7.
2 RELATED WORK

A market of heterogeneous users can often be divided into relatively more homogenous subgroups/segments of users that share similar characteristics. Differentiated services are thus provided \([46, 47]\). A typical example is the spot and on-demand services of Amazon EC2 that are for latency-critical and delay-tolerant jobs, respectively. The on-demand service has a fixed price with a small delay. For the spot service, users bid. A lower bidder will get served with a larger delay but pay a lower price. A CSP’s objective includes (i) revenue maximization \([9, 12–16]\), (ii) profit maximization \([4, 10, 11, 31]\), and (iii) social welfare maximization \([28–30, 32]\). When CSPs are public organizations or companies, social welfare maximization is their objective. Companies like Amazon and Microsoft are private where revenue and profit maximization is important; here profit is total revenue minus total cost. In marketing, market share is a key indicator of the market competitiveness of a CSP (i.e., how well a CSP is doing against its competitors), and it is the percentage of the total revenue in a market that a CSP’s business makes up \([50]\). For example, Gartner annually publishes the market shares of major CSPs \([1]\). Anyway, revenue management itself is also a branch of operations research \([51]\). The hybrid on-demand and spot services are mainly distinguished by the characteristics of their service models and the ways that servers are used to serve jobs (i.e., how queues are formed and jobs are processed). In the following, we introduce these related works.

Interruptible services are studied in References \([4, 9–15, 28–30, 32]\), while in this article and References \([16, 31]\) the offered services are non-interruptible, i.e., every job is processed continuously once started. Specifically, the first model of spot and on-demand services is proposed in References \([9, 10]\) and has been introduced in the last section. The realization of such service is defined by a preemptive priority queue. Differently, we explore two architectures for realizing it, since the realization with a non-preemptive priority queue performs poorly in our scenario. Our work is motivated by the situation that the on-demand service has well been accepted by plenty of users and its price is predefined, depending on not only user WTPs but also competition. We focus on service differentiation among such users and assume that users have the same initial WTP for the fastest on-demand service. The frameworks of References \([9, 10]\) works with the assumption that there are \(n\) classes of users whose initial WTPs are among \(n\) different values when the delay is zero; some users with low WTPs may not choose any service. In both frameworks, the WTP of a user decreases as the delay increases and the decreasing speed depends on its delay–cost type. Kilcioglu and Maglaras consider the setting in which the CSP has infinite servers like in Reference \([9]\). They specify the correlation (e.g., sublinear) between the WTP of each user under the minimal delay and its sensitivity to delay, and study the revenue improvement brought by the offering of spot service \([12]\).

The second service model focuses on enabling users to utilize the idleness of on-demand market. The idle periods of servers appear at random and are utilized as spot services by users who bid the highest \([24]\). Taking the effect of preemptions into account, Wu et al. show that the challenge of offering such spot service is to guarantee the immediacy of on-demand service and the persistence of spot service while sharing servers \([14]\). Then they give an integral resource allocation and pricing framework for this purpose, and it forms a DSIC mechanism. Further, an analytical expression is given to show the revenue improvement brought by such spot service whose viability is further illustrated numerically. Chen et al. consider two spot pricing schemes \([13]\). The first is a uniform discount scheme, and they derive the optimal discount price, given the customers’ expectation of the preemption probability. The second is an interruption-based discount scheme that provides customers with compensation for interruptions. The authors compare analytically or numerically the revenue improvements brought by the two schemes in different situations.
Interestingly, regardless of the architecture of servers to realize spot services, the viability of the interruptible spot service can also be studied by identifying factors that affect revenue and characterizing user’s sensitivity to market variability. Song and Guérin use probability distributions to characterize spot prices and job’s value and sensitivity to delay. They derive a condition under which a spot service is viable and give the optimal pricing and bidding strategies for a CSP and its users [15]. In this article and References [9–11], users all need to truthfully report their delay–cost types to the CSP. In contrast, in References [12, 15] spot prices are assumed to be drawn from a discrete set known by all users. Thus, each user’s optimal bid has a value from this set and does not directly disclose the value of its delay–cost type, although the bid is still related to its delay–cost type. Boodaghians et al. consider processing non-preemptive jobs on a single server [16]. Job parameters are assumed to be drawn from an underlying distribution known by the CSP. They design a truthful posted price mechanism that aims to maximize the revenue in expectation; here a user does not need to report its private information such as job value and delay requirement to the CSP explicitly.

Amazon EC2 is a major CSP. Understanding its internal pricing scheme is important in that its users can better know how to use spot services cost-effectively and other CSPs can get informed of the secret of Amazon EC2’s success, i.e., the way that Amazon generates revenue. Ben-Yehuda et al. study the statistical characteristics of the past spot prices in Amazon EC2 market and reveal that in practice Amazon probably sets the spot prices artificially to create uncertainty [6]. Finally, spot services are popular in that users can trade their delay tolerance for cheaper service. However, it is preemptive and creates significant complexity that users have to face [20, 25–27]. Tools have to be created to help users manage preemptions [21, 22], which establishes a barrier to the use of spot/delayed service.

There are many nice works that use auction theory to explore potential frameworks for pricing and managing computing resources. These frameworks take into account additional requirements of jobs such as soft or hard deadlines [4, 28–30] and virtual machine configuration [31, 32] where the availability of resource to a user depends on its bid and is unstable. In contrast to the above spot services and the delay-differentiated services of this article, these works do not consider an important business constraint in practice: Any CSP who wants to maintain its attractiveness must offer a standard on-demand service alongside any other offerings [11, 14], since, like what we do in our daily life, many users want a stable access to the computing service at a fixed unit price. Methodologically, these works [4, 28–32] use the theory of approximation algorithms and analyze the worst-case performances of their resource allocation and pricing schemes. This is different from our work and the above works [9–15], which analyze the expected performance of an ecosystem and some of which are based on queueing theory [9–11, 14].

Differently from all works above, Dierks and Seuken consider the objective of maximizing the utilization of servers [48]. They propose a heuristic server allocating policy that specifies a condition to judge whether each arriving job is admitted into the system and served. This policy needs knowledge of the variances of jobs. Finally, a pricing scheme is proposed to simply incentivize users to classify their jobs into different types where each type of jobs have similar variances. They show that the proposed variance-based policy can improve the utilization of servers significantly. In a two-provider market setting, the authors also perform a game-theoretic analysis of the resulting competitive effects [49].

3 A QOS-DIFFERENTIATED SERVICE MODEL

In this section, we describe the proposed QoS-differentiated service model and the associated questions to be addressed. The service model is generic, and we postpone the description of the ways of fulfilling its SLAs, which will be given after we study the model properties.
3.1 A QoS-Differentiated Service Model

Users arrive at the service system over time. Each user $j$ requests at time $a_j$ to continuously utilize a server for some time $s_j$. We equivalently refer to such a request as a job $j$, $a_j$ as arrival time, and $s_j$ as service time. Upon arrival, a job may get served with some delay $\phi$, i.e., it will get served at time $a_j + \phi$; then the service stops until the job is continuously served for a duration $s_j$. Once a user gets enough service time, it departs. The standard on-demand service in cloud markets represents the fastest service to satisfy all users. We use $T$ and $p$ to denote its delay and its price of utilizing one server per unit of time, and they are fixed exogenous parameters: $T$ is the minimum delay before a user can get served where $\phi \geq T$, and $p$ is the maximum price that a user need to pay for service.

3.1.1 Delay–Cost Function. The WTP of a user will decrease as the delay increases. Each user has an individual delay–cost type $\alpha$, which measures its sensitivity to the delay. For example, for a user of a larger $\alpha$, its WTP decreases faster as the delay increases. Generally, the users’ WTPs will be characterized by a family of functions, denoted by $u(\alpha, \phi)$.

Property 1. The WTP function $u(\alpha, \phi)$ is assumed to have the following properties where $\alpha$ is a positive real number and $\phi \in [T, +\infty)$:

(i) Normalisation: for all $\alpha \in R^+$, we have $u(\alpha, T) = p$;
(ii) Non-increasing: fixing the value of $\alpha$, $u(\alpha, \phi)$ is decreasing in $\phi$;
(iii) Monotone Parametrisation: fixing the value of $\phi$, $u(\alpha, \phi)$ is decreasing in $\alpha$;
(iv) Decreasing speed: fixing the value of $\phi$, $\frac{\partial u}{\partial \alpha}$ is decreasing in $\alpha$.

Given the specific form of $u(\alpha, \phi)$, each user will choose a specific delay–cost type $\alpha$ that can best fit its sensitivity to delay. The first subproperty implies that every user can accept on-demand service at a price $p$, since its WTP is $p$ when the delay is $T$. The second subproperty means that the WTP will decrease as the delay $\phi$ increases. The third subproperty states under the same delay $\phi$ that the larger the value of $\alpha$, the smaller the WTP $u(\alpha, \phi)$. Thus, when the delay increases from $T$ to a larger $\phi$, a user of larger $\alpha$ has more value loss and is more sensitive to delay. $\frac{\partial u}{\partial \phi}$ represents the slope of the tangent line at a point. The fourth subproperty guarantees that if a user has a larger $\alpha$, then the decreasing speed of its WTP is also larger.

For analytical tractability, the literature of cloud services [9–13, 15] follows the convention that the WTP functions are linear, where the WTP of a user decreases by $\alpha$ if it experiences one time unit of delay, to capture the phenomenon that a user’s WTP decreases as the delay increases; then, we have

$$u(\alpha, \phi) = p - \alpha \cdot (\phi - T), \quad \phi \in [T, +\infty).$$

In the extreme case that a user has $\alpha = 0$, it is insensitive to delay and its WTP is still $p$ even if it experiences an infinite delay. In the case that a user has $\alpha \to +\infty$, it is extremely sensitive to delay, and its WTP becomes negative even if it experiences a delay $\phi$ slightly larger than $T$. There may be another type of WTP functions that are concave [15], e.g.,

$$u(\alpha, \phi) = p - (\alpha \cdot (\phi - T))^3.$$  

Such functions may be useful for scenarios where the WTPs of delay-tolerant users decrease slowly before the delay increases to a threshold, after which users become very sensitive to delay and their WTPs decrease sharply [33]. For instance, Song and Guérin [15] give analytical results for the linear case like in References [9–13] while numerical results are also given for the concave case.

We thus propose Property 1 to generalize both linear and concave WTP functions and make our conclusions generic whenever possible. Except the results in Section 5.5, the theoretical results of this article hold if the WTP function satisfies Property 1. The linear WTP function allows us
to derive a closed-form expression of the performance of a SMS-based system in Section 5.5. The analytical results in the case of the linear WTP functions can serve as a guide to the case of the concave WTP functions, which is illustrated in Section 6.2.4. Even if the CSP offers a service system whose government is built on the assumption that the WTP functions of all users are linear, a user with a specific delay–cost type $\alpha$, whose WTP function satisfying Property 1 is actually concave, still has ways to participate in the service system. For example, its concave WTP function $u'$ can be lower bounded by a linear WTP function $u''$, i.e., $u''(\alpha, \varphi) \leq u'(\alpha, \varphi)$; here $u''$ is defined by two points of $u'$ in a two-dimensional space: $(T, p)$ and $(x, 0)$, where $u'(\alpha, x) = 0$. The user can use $u''$ to participate in the service system and trade its delay tolerance for cheaper services.

**Remarks.** In this article, the expected behaviors and performance of the system are analyzed. In reality, a user is typically an organization or company who engages in some long-term activities [2], and it is characterized by a set of features such as its delay–cost type. The user’s jobs are dynamically generated and submitted to the CSP over time. Users of the same features will have the same strategy to participate in the cloud service system and are treated by the CSP in the same way (e.g., assigned to the same service class, pay the same unit price). From the CSP’s perspective, a physical user with many jobs can be viewed as many (virtual) users, each of which has a single job and the same features as the physical user. Thus, theoretically, for ease of mathematical modeling and analysis, it is often assumed without loss of generality that one user represents one job in the literature [9–13, 15], while each user is maximizing its expected surplus.

3.1.2 Service Model. The CSP plans to offer $L$ SLAs to serve its users. For all $l \in [1, L]$, the $l$th SLA specifies a delay $\varphi_l$ and the price $p_l$ of utilizing a server per unit of time; for the users operating under the $l$th SLA, whenever their requests arrive, the CSP guarantees that the expected delay of delivering service is at most $\varphi_l$. The first SLA represents the standard on-demand service in cloud markets, and it is for latency-critical users who are not willing to tolerate significant delays. Thus, $p_1$ and $\varphi_1$ equal the price and delay of an on-demand service. The prices of the other SLAs are lower than $p_1$, at the expense of delaying the delivery of computing services to their consumers; here we let

$$T = \varphi_1 < \varphi_2 < \cdots < \varphi_L.$$  \hspace{1cm} (3)

Further, we have for all $l \in [1, L - 1]$ that the price of the $l$th SLA is larger than the price of the $(l + 1)$-th SLA; otherwise, users would prefer the $l$th SLA with a smaller delay. Thus, we have

$$p = p_1 > p_2 > \cdots > p_L.$$  \hspace{1cm} (4)

The interaction process between a CSP and its users is illustrated in Figure 1. Specifically, each user refers to the specific form of WTP functions $u(\alpha, \varphi)$ used by the CSP (e.g., (1)) and will choose a positive real number $\alpha \in \mathbb{R}^+$ such that $u(\alpha, \varphi)$ can best fit its sensitivity to delay; then, it reports the chosen $\alpha$ to the CSP. Users of the same $\alpha$ is said to have the same delay–cost type. The CSP aims to satisfy all its users, without rejecting any service request, since all users can accept the on-demand service. Under an arbitrary SLA $l \in [1, L]$, the surplus of a user is its WTP minus the SLA price, i.e., $u(\alpha, \varphi_l) - p_l$. According to the reported type, the CSP will choose one SLA for each type of users such that their surplus is maximized. Formally, we have the following definition.

**Definition 3.1.** A user of type $\alpha$ is assigned to the $l_\alpha$th SLA defined as follows:

$$l_\alpha = \arg \max_{l \in [1, L]} u(\alpha, \varphi_l) - p_l.$$  \hspace{1cm} (5)

The CSP regulates that if a user achieves the same maximum surplus under multiple SLAs, then it will be assigned to the SLA whose number is the largest.

ACM Trans. Model. Perform. Eval. Comput. Syst., Vol. 8, No. 3, Article 6. Publication date: June 2023.
Fig. 1. The interaction between users and a CSP.

Finally, we assume without loss generality that each SLA will be assigned a non-empty set of users. The case that a CSP offers \( L \) SLAs but there exist \( \zeta \) SLAs that are assigned no users is equivalent to the case that \( L - \zeta \) SLAs are offered and each SLA is assigned a non-empty set of users, where \( \zeta \leq L \). In fact, as we will see from the numerical results, the revenue of a CSP increases slightly as \( L \) increases.

### 3.2 System Objectives

The setting of jobs basically follows \[9, 10\]. An exception is that we consider the case that all users have the same WTP to accept the standard on-demand service. Thus, users are differentiated by their delay–cost types. As described in Section 3.1, each user corresponds to a job. Multiple users may have the same type. All types constitute a finite set \( \Phi \); the minimum and maximum values of the elements of \( \Phi \) are \( \alpha \) and \( \overline{\alpha} \). Let \( P(\alpha) \in (0, 1) \) denote the probability that an arriving user has a delay–cost type \( \alpha \), where

\[
\sum_{\alpha \in \Phi} P(\alpha) = 1. \tag{6}
\]

The mean arrival rate of the jobs of all types is \( \Lambda \), and the mean job size is \( s \).

Differently from References \[9, 10\], we need to determine which SLA a specific type of users belong to. A user’s WTP is dominated by its delay–cost type \( \alpha \). By Definition 3.1, its SLA assignment depends on its type \( \alpha \) and the prices and delays of the \( L \) SLAs. For all \( l \in [1, L] \), let \( \Phi_l \) denote the set of the types of users assigned to the \( l \)th SLA; under the assumption at the end of Section 3.1.2, we have

\[
\Phi_l \neq \emptyset. \tag{7}
\]

Let \( \mathcal{P} = \{ \Phi_1, \Phi_2, \ldots, \Phi_L \} \). Since each user can accept the standard on-demand service (i.e., the first SLA), it will be assigned to one of the \( L \) SLAs; thus, we have

\[
\bigcup_{l=1}^{L} \Phi_l = \Phi. \tag{8}
\]

By Definition 3.1, each user will be assigned to a single SLA and we have \( \Phi_{l_1} \cap \Phi_{l_2} = \emptyset \) for all \( l_1, l_2 \in [1, L] \) with \( l_1 \neq l_2 \). The mean job arrival rate of the \( l \)th SLA is

\[
\Lambda_l = \Lambda \cdot \sum_{\alpha \in \Phi_l} P(\alpha). \tag{9}
\]

For all \( l \in [1, L] \), the \( l \)th SLA guarantees that its jobs experiences a delay of at most \( \varphi_l \). The \( \mathcal{P} \) determines the job arrival rate of each SLA by Reference (9). Roughly, in a queueing system, the more the available servers, the smaller the expected delay of serving jobs. When there are \( x \) servers and \( \mathcal{P} \) is given, the expected delay \( t_l \) of the jobs of SLA \( l \in [1, L] \) is a non-increasing function of
Fig. 2. Exogenous and endogenous parameters: $x \rightarrow y$ denotes the dependence of the parameter $y$ on the parameter $x$; when it comes to the SMS architecture, additional parameters $\{m_l\}_{l=1}^L$ are involved, which will be introduced later in Section 5.1.3.

$x$. Suppose there are a total of $x = m$ servers for fulfilling all SLAs. The CSP will provide the minimum number $m$ of servers needed to fulfill SLAs such that

$$h(m, \mathcal{P}) = (t_1, t_2, \ldots, t_L) \leq (\varphi_1, \varphi_2, \ldots, \varphi_L). \quad (10)$$

We will leverage queueing theory to concretize the function $h(\cdot)$, which is elaborated in Section 5; see Equations (16) and (18). Exogenous parameters are parameters that are not affected by other variables in the system, while endogenous parameters are parameters that are influenced by other factors in the system. In Figure 2, we illustrate the main exogenous and endogenous parameters of this article. The relations implied by the blue arrows (respectively, the golden arrows) will be introduced in Section 4 (respectively, Section 5). Theoretically, given the set of the exogenous parameters in Figure 2, we will study in this article the way of determining the endogenous parameters and evaluate the performance of the proposed service mode. The total workload of users that is processed per unit of time under the $l$th SLA is $w_l = \Lambda \cdot s$. The revenue from the $l$th SLA per unit of time is $p_l \cdot w_l$. The total revenue obtained per unit of time is

$$G = \sum_{l=1}^L p_l \cdot w_l = \sum_{l=1}^L p_l \cdot \sum_{\alpha \in \Phi_l} P(\alpha) \cdot \Lambda \cdot s. \quad (11)$$

The CSP’s objective is to maximize the revenue (Equation (11)) while satisfying Equation (10). Its decision variables are those endogenous variables, including $\{p_l\}_{l=1}^L$, $\{\varphi_l\}_{l=1}^L$, and $\mathcal{P}$.

As illustrated in Figure 1, each user needs to report its type information to the CSP. However, this information is private, and users may seek possible ways to maximize their surplus by misreporting their type information. A mechanism is said to be DSIC if a user gains most or at least not less by being truthful, regardless of what the others do [23]. We will prove that the proposed service model is DSIC. In the context of this article, we have the following definition.

Definition 3.2. Every user of type $\alpha$ will report a type $\alpha'$ to the CSP, with the aim to maximize its surplus. Our service framework is said to be DSIC if the user’s surplus is maximized when it truthfully reports its type, i.e., $\alpha' = \alpha$, no matter whether the other users will truthfully do so.

Now, we have described the system objectives as far as the CSP and users are concerned. In the next section, we will first give some market properties in the aspects of SLA assignment $\mathcal{P}$, SLA prices and incentive compatibility. These results are independent of the way of fulfilling the SLAs.
Table 1. Key Notation

| Symbol | Explanation |
|--------|-------------|
| $L$    | the number of SLAs |
| $\varphi_l$ | the delay of the $l$th SLA |
| $p_l$ | the price of the $l$th SLA |
| $T$ | the delay of on-demand service where $\varphi_1 = T$ |
| $p$ | the price of on-demand service where $p_1 = p$ |
| $m$ | the total number of servers possessed by a CSP |
| $\Lambda$ | the total job arrival rate |
| $s$ | the mean job size |
| $\lambda_l$ | at a single server, the job arrival rate of the $l$th SLA |
| $\hat{\lambda}_l$ | at a single server, the total job arrival rate of the first $l$ SLAs |
| $\Phi$ | the set of the types of all users |
| $P(\alpha)$ | the probability that an arriving user/job has a delay–cost type $\alpha$ |
| $\bar{\alpha}$ (respectively, $\alpha$) | the maximum (respectively, minimum) type of $\Phi$ |
| $\Phi_l$ | the set of the types of the users who are assigned to the $l$th SLA |
| $P$ | the set $\{\Phi_1, \ldots, \Phi_L\}$ |
| $\hat{\alpha}_1, \ldots, \hat{\alpha}_{L+1}$ | a division of $\Phi$ used to define $\Phi_1, \ldots, \Phi_L$ by (12) |
| $t_l$ | the expected job delay of the $l$th SLA |
| $G$ | the revenue of the proposed delay-differentiated service model |

to satisfy Equation (10), which will be introduced in Section 5. The main notation of this article is summarized in Table 1.

4 MARKET PROPERTIES

In this section, we study the related market properties. Specifically, in Section 4.1, we show a structural property of the endogenous parameters $\{\Phi_l\}_{l=1}^L$ illustrated in Figure 2. In Section 4.2, we derive the form of the optimal SLA prices $\{p_l\}_{l=2}^L$, which relate to the market segmentation $\{\Phi_l\}_{l=1}^L$ and the SLA delays $\{\varphi_l\}_{l=2}^L$; the proposed service model is also shown to be DSIC. These results are the basis to leverage queueing theory to analyze the revenue and viability of the proposed service systems in the next section.

4.1 Structural Market Segmentation

In this subsection, we will show that there exists a segmentation of the interval $[\underline{\alpha}, \overline{\alpha}]$ into $L$ consecutive sub-intervals such that a user whose type value is in the $l$th sub-interval is assigned to the $l$th SLA where $l \in [1, L]$. In other words, while assigning users to SLAs, the classes of users are non-increasing in their type values.

4.1.1 Results. Suppose we are given an arbitrary setting of the SLA delays $\varphi_1, \varphi_2, \ldots, \varphi_L$ and prices $p_1, p_2, \ldots, p_L$ that satisfy Equations (3) and (4).

Proposition 4.1. There exists a sequence $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \in \Phi$ such that the $l$th SLA will be assigned the users of type $\alpha \in \Phi_l$, where $\underline{\alpha} = \hat{\alpha}_{L+1} < \cdots < \hat{\alpha}_2 < \hat{\alpha}_1 = \overline{\alpha}$ and $\Phi_l$ is a subset of user types defined below:

$$
\Phi_l = \begin{cases} 
\Phi \cap (\hat{\alpha}_{l+1}, \hat{\alpha}_l], & \text{if } l \in [1, L-1], \\
\Phi \cap [\hat{\alpha}_{L+1}, \hat{\alpha}_L], & \text{if } l = L.
\end{cases}
$$
Proposition 4.1 is illustrated in Figure 3. The sequence \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \) defines a particular assignment of user types to SLAs, i.e., \( \Phi_1, \Phi_2, \ldots, \Phi_L \). In the rest of this article, we will simply call such a sequence as a specific market segmentation. By Proposition 4.1, while maximizing the revenue (Equation (11)), the endogenous variable \( \mathcal{P} = \{\Phi_1, \Phi_2, \ldots, \Phi_L\} \) is transformed into the market segmentation \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \).

4.1.2 Proof Outline. Now we provide an overview of the proof of Proposition 4.1. First, a user’s sensitivity to delay determines the degree of WTP reduction as the delay increases. Intuitively, if a user is more sensitive to delay, then its WTP will decrease more while facing the same increment in delay. Formally, we have the following relation.

**Lemma 4.2.** Let us consider two arbitrary users of types \( \alpha_1 \) and \( \alpha_2 \) with \( \alpha_1 > \alpha_2 \), and two SLAs \( k_1 \) and \( k_2 \) with \( k_1 < k_2 \). The user of type \( \alpha_1 \) is more sensitive to delay as explained for Property 1; the SLA delays satisfy \( \varphi_{k_1} < \varphi_{k_2} \) by (3). Then, we have that the difference of the WTPs of the user of type \( \alpha_1 \) under the \( k_1 \)-th and \( k_2 \)-th SLAs is larger than its counterpart for the user of type \( \alpha_2 \), i.e.,

\[
 u(\alpha_1, \varphi_{k_1}) - u(\alpha_1, \varphi_{k_2}) > u(\alpha_2, \varphi_{k_1}) - u(\alpha_2, \varphi_{k_2}).
\]

Second, we show an orderly pattern in SLA assignment. By Definition 3.1, each user is assigned to a SLA where its surplus is maximized. Driven by Lemma 4.2, we can prove that a user of larger \( \alpha \) is more sensitive to delay and will be assigned to a SLA with a smaller delay. Formally, we have the following conclusion.

**Lemma 4.3.** Let us consider two users of types \( \alpha_1 \) and \( \alpha_2 \) where \( \alpha_1 > \alpha_2 \). If the users of types \( \alpha_1 \) and \( \alpha_2 \) are respectively assigned to the SLAs \( k_1 \) and \( k_2 \) (i.e., \( \alpha_1 \in \Phi_{k_1} \) and \( \alpha_2 \in \Phi_{k_2} \)), then we have that \( k_1 \leq k_2 \), where the SLA delays satisfy \( \varphi_{k_1} \leq \varphi_{k_2} \) by (3).

Finally, with the assumption in Section 3.1.2, each SLA will be assigned a non-empty set of user types, i.e., \( \Phi_l \neq \emptyset \) for all \( l \in [1, L] \). For two SLAs \( k_1 \) and \( k_2 \), if \( k_1 < k_2 \), then we have by Lemma 4.3 that the type value in \( \Phi_{k_1} \) is larger than the type value in \( \Phi_{k_2} \). Since each user can accept the first SLA with a non-negative surplus, no users will be rejected and each user will finally be assigned to a specific SLA. We can thus derive Proposition 4.1. To summarise, Proposition 4.1 is built on Lemma 4.3, which is based on Lemma 4.2. See Appendix A.1 for their detailed proofs.

4.2 Optimal DSIC Mechanism

By Proposition 4.1, there is a structural market segmentation. In this subsection, suppose we are given an arbitrary market segmentation \( \{\Phi_l\}_{l=1}^L \) and SLA delays \( \{\varphi_l\}_{l=1}^L \). Then, we derive the corresponding optimal SLA prices \( \{p_l\}_{l=2}^L \) to maximize the revenue (11).

4.2.1 Results. Below, we use the predefined market segmentation \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \) and SLA delays \( \varphi_1, \varphi_2, \ldots, \varphi_L \) to define \( L \) parameters \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \) that will be used to define SLA prices.

**Definition 4.4.** Let \( u_l^- = u(\hat{\alpha}_l, \varphi_{l-1}) - u(\hat{\alpha}_l, \varphi_l) \) for all \( l \in [2, L] \) where \( u_l^- \) is the difference of the WTPs of a user of type \( \hat{\alpha}_l \), respectively, under the \((l-1)\)-th and \(l\)-th SLAs. We define parameter \( \hat{p}_l \) to
be such that (i) \( \hat{p}_1 = u(\hat{\alpha}_1, \varphi_1) = \hat{p} \), i.e., the price of on-demand instances, and (ii) for all \( l \in [2, L] \), \( \hat{p}_l \) is the maximum possible \( p_l \) that satisfies \( p_l \leq \hat{p}_{l-1} - u_l \), i.e., \( \hat{p}_l = \hat{p}_l(\hat{\alpha}_1, \ldots, \hat{\alpha}_l, \varphi_1, \ldots, \varphi_l) = \hat{p}_{l-1} - u_l = \hat{p}_1 - \sum_{l'=2}^l u_{l'} \).

The SLA prices, the SLA delays and the market segmentation are three correlated market features, and one feature changes as the others change. Given the set of user types \( \Phi \), the resultant market segmentation relates to the particular SLAs prices and delays according to Definition 3.1. When the SLA prices are set to \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \), the resultant market segmentation depends on these newly given SLA prices and the predefined SLA delays, and we denote it by \( \hat{\alpha}_1', \hat{\alpha}_2', \ldots, \hat{\alpha}_{L+1}' \). We need to verify the consistency between the predefined and resultant market segmentations and check whether \( \hat{\alpha}_l' = \hat{\alpha}_l \) for all \( l \in [1, L] \) or not. For all \( l \in [1, L] \), we still relate \( \Phi_l \) to \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \), and \( \Phi_l \) is given in Equation (12). The following proposition shows the consistency.

**Proposition 4.5.** Suppose the SLA prices are set to \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \) and the SLA delays are the predefined \( \varphi_1, \varphi_2, \ldots, \varphi_L \). For all \( l \in [1, L] \), the users of type \( \alpha \in \Phi_l \) are assigned to the \( l \)th SLA, i.e., the resultant market segmentation is still \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \).

The following conclusion is a main result of this section.

**Proposition 4.6.** Suppose we are given an arbitrary market segmentation \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \), and the SLA delays \( \varphi_1, \varphi_2, \ldots, \varphi_L \). Then, we have that (i) the proposed service model is DSIC, and (ii) when the SLA prices \( p_1, p_2, \ldots, p_L \) are set to \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \), we have that \( p_1, p_2, \ldots, p_L \) are the optimal SLA prices to maximize the CSP’s revenue.

If two SLAs are offered, then we have by Proposition 4.6 that the price of the second SLA is \( p_2 = \hat{p}_2 = u(\hat{\alpha}_2, \varphi_2) \), which is also illustrated in Figure 3. The users of the second SLA are more delay-tolerant and can accept more delay. \( 1 - \frac{u(\hat{\alpha}_2, \varphi_2)}{\hat{p}} \) is the discount offered to the users. For example, suppose \( p = 1, \hat{\alpha}_2 = 4, \) and \( \varphi_2 - T = 0.1 \); then, if the WTP function is (1), \( u(\hat{\alpha}_2, \varphi_2) = 0.6 \) and the discount is 40%. Proposition 4.6 is independent of the architectures of servers used for fulfilling SLAs and provides the relation of the SLA prices \( \{p_l\}_{l=1}^L \) to the market segmentation \( \{\hat{\alpha}_l\}_{l=1}^{L+1} \) and the SLA delays \( \{\varphi_l\}_{l=1}^L \).

4.2.2 Proof Overview. Now we give an overview of the proofs of Propositions 4.5 and 4.6. The detailed proofs can be found in Appendix A.2. To prove Proposition 4.5, we first consider the surpluses of a user of type \( \alpha \in \Phi_l \) under two adjacent SLAs whose numbers are simultaneously no larger or smaller than \( l \). Roughly, its surplus under the SLA whose number is closer to \( l \) is always larger, as shown below.

**Lemma 4.7.** Suppose the SLA prices are set to \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \) and the SLA delays are the predefined \( \varphi_1, \varphi_2, \ldots, \varphi_L \). Given two arbitrary \( l, l' \in [1, L] \), let us consider a user of type \( \alpha \in \Phi_l \) and its surplus under the \( l' \)th SLA. The surplus of this user is such that (i) in the case that \( l' \in [2, L] \),

- if \( \alpha = \hat{\alpha}_1 \) and \( l' = l \), then its surpluses under the \( l' \)th and \( (l' - 1) \) th SLAs are the same;
- otherwise, its surplus under the \( l' \)th SLA is larger than its surplus under the \( (l' - 1) \)th SLA;

and (ii) in the case that \( l' \in [1, L - 1] \), its surplus under the \( l' \)th SLA is larger than its surplus under the \( (l' + 1) \) th SLA.

See Appendix A.2 for the proof of Lemma 4.7. By Lemma 4.7, we can use the transitivity of inequalities to derive for all \( l \in [1, L] \) that the user of type \( \alpha \in \Phi_l \) achieves the maximum surplus under the \( l \)th SLA and will thus be assigned to the \( l \)th SLA when the SLA prices are set to \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \) and the SLA delays are the predefined \( \varphi_1, \varphi_2, \ldots, \varphi_L \). We can thus prove Proposition 4.5 and derive that the resultant market segmentation equals the predefined market segmentation.
The idea in proving Proposition 4.6 is as follows. In our service model, a nice property is that it always assigns each user to a SLA under which the user’s surplus is maximized. Naturally, when a user misreports its type, it may be assigned to a SLA under which the user gains a smaller surplus. Thus, users have no willingness to misreport and our service framework is DSIC. However, given the market segmentation \( \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{L+1} \), all users of type \( \alpha \in \Phi_l \) will be assigned to the \( l \)th SLA where \( l \in [1, L] \). We can show that \( \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_L \) are the maximum possible prices to guarantee this, and they are thus optimal.

5 SUPPORTING ARCHITECTURES AND PERFORMANCE

In Sections 3 and 4, we study a generic service model that offers \( L \) SLAs and its pricing properties. The SLA fulfillment relies on proper provision of servers to jobs to satisfy Equation (10). In Section 5.1, we will introduce the PBS and SMS architectures, respectively, which are two candidate server architectures used to serve jobs and fulfill SLAs; here we also give the relations between the endogenous parameters \( \{ \Lambda_l \}_{l=1}^L \), \( \{ t_l \}_{l=1}^L \), and \( \{ m_l \}_{l=1}^L \) illustrated in Figure 2. The service model together with each architecture forms an integral service system. In Section 5.2, we give the baseline used to show the performance of the proposed service systems. In Section 5.3, we formally derive the relation between \( \{ t_l \}_{l=1}^L \) and \( \{ \phi_l \}_{l=1}^L \) while maximizing a CSP’s revenue.

In the rest of this section, we present the main results of this article. Specifically, we give a performance bound in Section 5.4, which somewhat reveals the inviability of the PBS-based service system. For the SMS-based service system, we give in Section 5.5 a closed-form expression of its revenue improvement on a pure on-demand service system when two SLAs are offered; here numerical results are also accompanied to illustrate that the SMS-based service system can achieve a significant revenue improvement under a wide range of conditions. Finally, when more than two SLAs are offered, an exhaustive search procedure is given in Section 5.6 to determine the optimal SLA prices and delays, which allows for giving more numerical results in the next section to illustrate the performance of the SMS-based service system.

5.1 Two Supporting Architectures

Now, we introduce the two architectures. A CSP has a total of \( m \) servers. When a job \( j \) arrives, it is assigned to a server that will serve it for a duration \( s_j \). We will respectively consider (i) the PBS architecture and (ii) the SMS architecture. In the former, an arriving job will be assigned to one of the \( m \) servers, and the order of serving the jobs at a server depends on their priorities, which depend on the SLAs to which they belong. In the latter, servers are separated into \( L \) groups and each exclusively serves the jobs of the same SLA. We leverage queueing theory to give the relation between the expected delay \( t_l \) of each SLA \( l \in [1, L] \) and the job arrival rate \( \Lambda_l \).

5.1.1 Preliminary. The performance of cloud service systems closely relates to the particular policy that assigns arriving jobs to servers. Before elaborating the architectures, we first introduce these policies. Suppose there are \( m' \) servers to serve a particular group of jobs and the mean job arrival rate is \( \Lambda' \). Typical dispatching policies include (i) Random, where for every job it chooses every server with the same probability \( \frac{1}{m'} \) and assigns \( j \) to the chosen server [34, 35], and (ii) Round-Robin (RR), where jobs are assigned to servers in a cyclical fashion with the \( j \)th job being assigned to the \( i \)th server where \( i = j \bmod m' \) [36]. Under both policies, jobs are evenly dispatched over the \( m' \) servers. At each server, the arriving jobs form a single queue with the same mean job arrival rate \( \lambda' = \frac{\Lambda'}{m'} \) [37]. The service time of a job is denoted by a random variable \( X \), and the mean \( s \) of \( X \) is normalized to be 1, i.e., \( s = 1 \).

In practice, the RR and Random policies are supported by Amazon EC2 to dispatch jobs to servers while on-demand users are being served [38]. Their prevalence is due to the following reasons.
Each job needs an individual job assignment decision. Such policies do not need the knowledge of server states and can form a distributed scheduler where numerous job assignment decisions could be done instantaneously, thus reducing the scheduling delays. However, the maintenance of the state information of all servers relies on a heartbeat mechanism where servers communicate on their states with a centralized scheduler at a specific frequency and the job assignment decisions are also made at such a frequency [39]. In large-scale cloud server systems, to reduce communication overhead, the frequency has to be low, which leads to that the scheduling delay is too large [18, 40, 41]. Thus, a distributed scheduler is used in cloud service systems.

5.1.2 The PBS Architecture. In the PBS architecture, whenever a job arrives, it is assigned to one of the $m$ servers by some dispatching policy, as illustrated in Figure 4. The total job arrival rate is $\Lambda$, and the job arrival rate at a single server is

$$\lambda = \frac{\Lambda}{m} \in (0, 1).$$  \hspace{1cm} (13)

At every server, the jobs have $L$ priority classes. For all $l \in [1, L-1]$, the jobs of SLA $l$ have higher priority to utilize servers than the jobs of SLAs $l+1$ and are said to have a priority $l$. At the moments of job completion, the server becomes idle and will select a new job of the highest priority to serve, and jobs of the same priority will be chosen in a first-come-first-served (FCFS) discipline. While a job $j$ is being served, the nonpreemptive rule is applied, that is, the job will continuously occupy a server for a duration $s_j$ even if other jobs of higher priorities arrive.

Now, we give the mean delay $t_l$ of the jobs of each SLA $l \in [1, L]$. By Equations (7) and (9), at each server, the job arrival rate of the $l$th SLA is

$$\lambda_l = \Lambda \cdot \sum_{\alpha \in \Phi_l} P(\alpha) / m = \lambda \cdot \sum_{\alpha \in \Phi_l} P(\alpha) > 0.$$  \hspace{1cm} (14)

The total arrival rate of the jobs of SLAs $1, \ldots, l$ is

$$\hat{\lambda}_l = \sum_{l'=1}^{l} \lambda_{l'}$$ where $\hat{\lambda}_L = \lambda.$ \hspace{1cm} (15)

The jobs of all SLAs at every server form a single queue and their job arrivals are described as a Poisson process with rate $\lambda$. The service time $X$ of jobs is assumed to follow a general distribution where the mean $s$ is 1. Such a queue is usually denoted by $M/G/1$. We can directly use the result
Fig. 5. The separated multi-SLAs architecture with $L = 2$ and $m_1 + m_2 = m$: Colored rectangles denote jobs of different SLAs while colored circles denote servers of different SLAs.

for a $M/G/1$ queue with priority in Reference [3] and get the delay of the jobs of the $l$th SLA,

$$t_l = \frac{0.5 \cdot \lambda \cdot E[X^2]}{(1 - \hat{\lambda}_{l-1}) \cdot (1 - \hat{\lambda}_l)},$$

(16)

where $l \in [1, L]$, $\hat{\lambda}_0$ is set to zero trivially, and $E[X^2]$ is the second moment of $X$, i.e., its meansquared value.

5.1.3 The SMS Architecture. In the SMS architecture, the $m$ servers are separated into $L$ groups, and each group has $m_l$ servers, where

$$m = \sum_{l=1}^{L} m_l.$$  

(17)

Differently from the PBS architecture, there are $L$ additional endogenous variables $\{m_l\}_{l=1}^{L}$, whose optimal values need to be determined subsequently. The $l$th group is used to exclusively serve the jobs of the $l$th SLA, and every job that belongs to the $l$th SLA will be assigned to one of the $m_l$ servers under some dispatching policy such as Random or RR, as illustrated in Figure 5. At every server, the jobs will be served in a FCFS discipline. By Equation (9), the total job arrival rate of the $l$th SLA is $\Lambda_l$; then the job arrival rate at a single server is $\lambda_l = \frac{\Lambda_l}{m_l}$. The jobs at every server forms a single queue, and when it is a $M/G/1$ queue, we have from Reference [3] that the job delay of the $l$th SLA is

$$t_l = 0.5 \cdot \lambda_l \cdot E[X^2]/(1 - \lambda_l).$$

(18)

Beyond the above architectural description, we will use in this article an exponential, hyperexponential, or heavy-tailed distribution to model the service time $X$. As often used in cloud and server systems [34, 40, 55, 56], they have available closed-form expressions for $E[X^2]$ and can guarantee the existence of $E[X^2]$, which enable analytically evaluating the performance of the architectures above. When $X$ follows an exponential distribution [34, 40], we have

$$E[X^2] = 2 \cdot s^2 = 2.$$  

(19)

When $X$ follows a hyperexponential distribution [40], it can be characterized by $h$ tuples $(\pi_i, \eta_i)$, where $i \in [1, h]$ and $\sum_{i=1}^{h} \eta_i = 1$: $X$ has a probability $\eta_i$ to follow an exponential distribution with
rate $\pi_i$. For an exponential distribution with rate $\pi_i$, its mean is $\frac{1}{\pi_i}$. The mean of $X$ is

$$s = \sum_{i=1}^{h} \frac{\eta_i}{\pi_i} = 1,$$  \hspace{1cm} (20)

and the second moment of $X$ is

$$E[X^2] = \sum_{i=1}^{h} \frac{2}{\pi_i^2} \cdot \eta_i.$$  \hspace{1cm} (21)

When $X$ follows a Pareto distribution that is a type of heavy-tailed distribution [34, 41, 54], we consider the case where all jobs have finite sizes and the second moment of $X$ exists (i.e., $a > 2$). Then, we have

$$E[X^2] = \frac{a \cdot x_m^2}{a - 2} = \frac{(a - 1)^2}{a^2 - 2 \cdot a} > 1 \quad \text{if} \quad a > 2,$$  \hspace{1cm} (22)

where $a$ is the shape parameter and $x_m \in (0, 1)$ is the scale parameter when $E[X] = \frac{a \cdot x_m}{a - 1} = 1$.

When $X$ follows a log-normal distribution that is also a type of heavy-tailed distribution [54–56], the second moment of $X$ always exists and we have

$$E[X^2] = e^{2b + 2c^2} = e^{c^2} > 1,$$  \hspace{1cm} (23)

where $b$ and $c > 0$ are two parameters, $E[X] = e^{b + \frac{c^2}{2}} = 1$, and the variance is $e^{c^2} - 1$.

5.2 Benchmark: The Standard On-demand Service System

The delay-differentiated service system of this article can be viewed as a complement to the standard on-demand service model, which will be used as a benchmark. In a pure on-demand system, all jobs are served with a short delay and processed with the same priority on the $m$ servers. Upon arrival of each job, it will be dispatched to one of the $m$ servers under some policy and the jobs at the same server will be served in a FCFS discipline. The total job arrival rate is $\Lambda_{od}$, and the job arrival rate at a single server is $\lambda_{od} = \frac{\Lambda_{od}}{m}$. Similarly to Equation (18), we have that the delay of all jobs is

$$t = 0.5 \cdot \lambda_{od} \cdot E[X^2]/(1 - \lambda_{od}).$$  \hspace{1cm} (24)

The job delay will be no larger than $T$, which requires that $t \leq T$.

We denote by $G_{od}$ the maximum revenue that an on-demand service system can achieve when a CSP has $m$ servers. The on-demand service has a fixed price $p$ and guarantees a small delay of at most $T$. A CSP’s revenue is maximized when the delay is $T$ and we have by Equation (24) that the corresponding job arrival rate at a single server is as follows:

$$\lambda_{od} = T/(A + T),$$  \hspace{1cm} (25)

where $A = 0.5 \cdot E[X^2]$. Further, we have

$$G_{od} = m \cdot p \cdot \lambda_{od} \cdot s = m \cdot p \cdot T/(A + T).$$  \hspace{1cm} (26)

Let $G$ denote the revenue of our delay-differentiated service system. The viability of our service system is mainly indicated by the ratio of $G$ to $G_{od}$, denoted by $\gamma$:

$$\gamma = G/G_{od}.$$  \hspace{1cm} (27)

If $\gamma > 1$, then our service system achieves a higher revenue than the on-demand service system; the larger the value of $\gamma$, the higher the revenue improvement. $\gamma - 1$ represents how much the
revenue is improved by when our service system is used. For example, if \( \gamma = 1.5 \), then it means that the revenue is improved by 50%.

### 5.3 Optimal SLA Delays

Suppose there are \( m \) servers and the aggregate job arrival rate is \( \Lambda \). For all \( l \in [1, L] \), the arrival rate of jobs bound to the \( l \)th SLA is \( \Lambda_l \). In the PBS architecture, the \( m \) servers are shared among the jobs under all SLAs. In the SMS architecture, the servers are divided into \( L \) groups and the jobs under the \( l \)th SLA are assigned \( m_l \) servers to process its jobs, where \( \sum_{l=1}^{L} m_l = m \). For our delay-differentiated service system, no matter which architecture in Section 5.1 is used, the expected job delay under the \( l \)th SLA type is denoted by \( t_l \) and determined by Equation (16) or Equation (18).

As described in Equation (10), the expected delay \( t_l \) of the jobs of the \( l \)th SLA should be no larger than the delay \( \varphi_l \) that this SLA defines. The first SLA represents the standard on-demand service and its SLA delay is \( T \). Intuitively, we should keep the other SLA delays as small as possible, i.e., \( \varphi_l = t_l \) for all \( l \in [2, L] \), to maximize the revenue. In fact, by doing so, we can make every SLA price as high as possible, which can be proved by analyzing the structure of the SLA prices in Definition 4.4. Formally, the SLA delays relate to the expected job delays in the following way.

**Proposition 5.1.** To maximize the revenue of a CSP, we have for all \( l \in [2, L] \) that the SLA delay \( \varphi_l \) should be the expected delay \( t_l \) of the jobs of the \( l \)th SLA, no matter which supporting architecture is used.

See Appendix B for the proof of Proposition 5.1. With Proposition 5.1, we will further analyze in the rest of this section the performance of our service system respectively under the PBS and SMS architectures.

### 5.4 A Performance Bound of the PBS-based Service System

In this subsection, we will study the performance of the proposed service system when it is built on the PBS architecture. Priority queues are common in literature. It is desirable to show the viability of a PBS-based service system before we turn to study other architectures. When it comes to the PBS architecture, we use \( G_{obs} \) to denote the revenue of our delay-differentiated service system. As described in Equation (27), \( \gamma \) is the ratio of \( G_{pbs} \) to \( G_{od} \). We will get an upper bound of \( \gamma \) that is close to 1. This implies that, at best, it can marginally outperform the on-demand service system, which will discourage the adoption of a PBS-based service system.

**Proposition 5.2.** The performance of a PBS-based service model is upper bounded by \( 1 + T \Lambda \) times the optimal performance of the standard on-demand service model, in terms of the revenue, where \( \Lambda = 0.5 \cdot E[X^2] \).

See Appendix C.1 for the proof of Proposition 5.2. Now we illustrate the intuition in this proof. For example, when the service time \( X \) follows an exponential distribution, we have \( E[X^2] = 2 \) by Equation (19). For the first SLA, we have \( \hat{\lambda}_1 \in (0, 1) \) by Equations (13), (14), and (15); by Equation (10), the constraint that the delay of the first SLA is no larger than \( T \), where \( t_1 \leq T \), requires that \( \lambda < T \), which is due to Equation (16). This means that the total load \( \lambda \) of a server is low and the performance of a PBS-based service system is poor, since \( T \) is small.

The bound in Proposition 5.2 is only related to the delay \( T \) of the first SLA and the second moment \( E[X^2] \) of the job service time distribution. As mentioned at the end of Section 5.1, there are four typical distributions to model the job service time in the literature for cloud and server systems [34, 40, 41, 55, 56]. Below, we use these distributions to numerically illustrate the performance bound in Proposition 5.2.

When \( X \) follows an exponential distribution, we have \( \Lambda = 1 \) by Equation (19). When \( X \) follows a hyperexponential distribution, we use an example in Reference [40] to set \( h = 2, \eta_1 = 0.75 \).
\( \eta_2 = 0.25; \) we let \( \pi_1 \in (0, 1) \), which represents more jobs have relatively smaller service times. Given the value of \( \pi_1 \), we can get the value of \( \eta_2 \) by Equation (20) and further the value of \( A \) by (21), which is illustrated in Appendix C.2; here we have \( A > 1 \). When \( X \) follows a Pareto distribution with \( a > 2 \) or a log-normal distribution, we have \( A > 0.5 \) by Equations (22) and (23). To summarise, we have that, the upper bound in Proposition 5.2 is at most \( 1 + T \) when \( X \) follows an exponential or hyperexponential distribution and at most \( 1 + 2 \cdot T \) when \( X \) follows a Pareto distribution with \( a > 2 \) or a log-normal distribution. Thus, the PBS-based service system can only outperform the standard on-demand service system marginally, since the delay of the first SLA \( T \) is small. In the rest of this article, we will solely focus on studying the SMS-based service system alone.

### 5.5 A Closed-form Result for the SMS-based System with Two SLAs

In this section, we provide under mild assumptions a closed-form expression of the revenue improvement of a SMS-based service system over a pure on-demand service system, in the case of \( L = 2 \). Even in this simple case, the revenue improvement is remarkable and comparable to its counterpart in the case of offering more SLAs, as shown later. A specific setting of the service system can be defined by parameters such as the number of possessed servers, and the job arrival rate and the population’s delay tolerance. A direct result of the closed-form expression is an operational region of these parameters in which the SMS-based system can always lead to an increment in the revenue. Given a specific setting, one can also compute the optimal revenue improvement and system configuration easily.

**Assumptions.** In cloud markets, both the amount of servers and the population size are large so that the revenue from a single server or user could be negligible, in comparison with the total revenue. To obtain analytical results, we relax in this subsection the constraint that the number of servers assigned to each SLA is integer and allow the number to be fractional; we also allow the distribution of user types to be continuous. The total revenue after relaxation approximates the total revenue of an integer solution, which can be testified later: Figure 7(b) gives the optimal revenue improvement \( \gamma^* \) computed by the closed-form results here; Figure 10(a) contains the values of \( \gamma^* \) in the case of \( L = 2 \), which are computed by an exhaustive search procedure in the next subsection that gives an optimal solution for the discrete case.

In computing services, some typical conventions are followed for analytical tractability. Specifically, the WTP functions are linear and satisfy Equation (1) [9, 10, 15]. A uniform distribution is often used to model users’ valuations and delay tolerances so that each segment of users can well be represented in the system [10, 14, 15, 41, 42]. The price of on-demand service is normalized to be 1, i.e., \( p = 1 \). Given a delay–cost type \( \alpha \), let

\[
\varphi_z^\prime = \frac{1}{\alpha},
\]

and we have by Equation (1) that a user’s WTP becomes zero when the delay reaches \( \varphi_z = \varphi_z^\prime + T \); each type \( \alpha \) corresponds to a unique \( \varphi_z^\prime \), which is referred to as the relative zero-WTP point. For each arriving job, we assume that its relative zero-WTP point \( \varphi_z^\prime \) follows a continuous uniform distribution over \((0, \tau)\), which is equivalent to defining the probability distribution \( P \) of Section 3.2. For the jobs with \( \varphi_z^\prime \rightarrow 0 \), they cannot tolerate any delay larger than \( T \), since their WTP would become negative. The value of \( \tau \) represents the overall delay-tolerance level of the population. The lower and upper bounds of \( \alpha \) are \( \underline{\alpha} \) and \( \overline{\alpha} \). Correspondingly, we have \( \underline{\alpha} = 1/\tau \) and \( \overline{\alpha} \rightarrow +\infty \), and the user types are distributed over \([1/\tau, +\infty)\) to include both delay-sensitive and delay-tolerant jobs.

The on-demand service represents the fastest service and \( T \) denotes the delay to deliver service and is a system parameter. \( m \) is the amount of servers that a CSP has. The external market condition is defined by \( \Lambda \) and \( \tau \); here, \( \Lambda \) is the total arrival rate of all jobs with different delay tolerances. We call the tuple of these parameters \((T, m, \Lambda, \tau)\) as a system setting. Under a specific setting, the CSP
needs to determine the optimal proportion of jobs and the optimal number \( m_1 \) of servers assigned to each SLA \( l \in [1, L] \).

**Problem Formulation.** By Proposition 4.1, there exists a partition point \( \lambda_2 = y \) such that all users of type \( \alpha \in (y, \bar{y}] \) are assigned to the first SLA while the others of type \( \alpha \in [y, \bar{y}] \) are assigned to the second SLA where \( L = 2 \). Herein, \( y \) defines the market segmentation \( \{ \alpha_l \}_{l=1}^{L+1} : \Phi_1 = (y, \bar{y}] \) and \( \Phi_2 = [y, \bar{y}] \). Correspondingly, let \( x = 1/y \), denoting a relative zero-WTP point in \( (0, \tau] \); the jobs whose relative zero-WTP points are in \( (0, x) \) are assigned to the first SLA while the other jobs are assigned to the second SLA. The mean job arrival rate of all users is \( \Lambda \). Since the relative zero-WTP point of each arriving job is uniformly distributed over \( (0, \tau] \), we have like Equation (9) that the job arrival rates for the first and second SLAs are, respectively,

\[
\Lambda_1 = \Lambda \cdot x/\tau \quad \text{and} \quad \Lambda_2 = \Lambda \cdot (\tau - x)/\tau,
\]

where \( \Lambda = \Lambda_1 + \Lambda_2 \). Under a system setting, our decision variable is \( x \) and the revenue that a CSP obtains per unit of time is denoted by \( G(x) \) and given later in Equation (37). We focus on the system performance in the case that

\[
x \in (0, \tau) \quad \text{and} \quad \lambda_{od} \cdot m < \Lambda < m.
\]

By Equation (28), the former guarantees a positive job arrival rate for each SLA; the latter says that the total job arrival rate in a SMS-based service system is larger than the one in an on-demand service system but does not exceed the processing capacity of the \( m \) servers. By Equations (26) and (27), the revenue improvement of a SMS-based system over the on-demand service system is as follows:

\[
\gamma(x) = G(x)/G_{od} = (A + T) \cdot G(x)/(T \cdot m).
\]

For our SMS-based service system, we will seek for the optimal \( x^* \) under which the optimal revenue \( G(x^*) \) is achieved, which will be a function of the setting \( (T, m, \Lambda, \tau) \). Under the optimal \( x^* \), we make the following definition.

**Definition 5.3.** We define a feasible operational region as a condition of the system setting \( (T, m, \Lambda, \tau) \) under which the SMS-based service system improves the revenue of a CSP over the standard on-demand service system. In other words, in the feasible operational region, we have \( \gamma(x^*) > 1 \).

**System Configuration.** Now, we derive the relation of the endogenous variables \( \phi_2, \ p_3 \), and \( \{ m_l \}_{l=1}^{L} \) to the decision variable \( x \). For the first SLA, its SLA delay is an exogenous parameter \( T \), and we have by Equation (18) and (25) that the job arrival rate per server is

\[
\lambda_1 = \lambda_{od} = T/(A + T).
\]

The number \( m_1 \) of servers assigned to the first SLA is determined by \( \lambda_1 \) and \( \Lambda \); then we have

\[
m_1 = \frac{\Lambda_1}{\lambda_1} = \frac{\Lambda \cdot (A + T)}{\tau \cdot T} \cdot x \quad \text{and} \quad m_2 = \frac{\Lambda \cdot (A + T)}{\tau \cdot T} \cdot x,
\]

where equality (a) is due to Equation (28) and (b) is due to Equation (17). Regarding the second SLA, the job arrival rate per server is

\[
\lambda_2 = \frac{\Lambda_2}{m_2} = \frac{T \cdot \Lambda \cdot (\tau - x)}{\tau \cdot T \cdot m - \Lambda \cdot (A + T) \cdot x}.
\]
where equality (a) is due to Equations (28) and (32); the SLA delay is as follows:
\[
\varphi_2 \overset{(b)}{=} t_2 \overset{(c)}{=} A \cdot \lambda_2 \div (1 - \lambda_2) = \frac{A \cdot T \cdot \Lambda \cdot (\tau - x)}{\tau \cdot T \cdot m - \Lambda \cdot (A + T) \cdot x} \div \frac{\tau \cdot T \cdot (m - \Lambda) - \Lambda \cdot A \cdot x}{\tau \cdot T \cdot m - \Lambda \cdot (A + T) \cdot x}
\]
\[
= \frac{A \cdot \Lambda \cdot (\tau - x)}{\tau \cdot T \cdot (m - \Lambda) - \frac{\Lambda \cdot A \cdot x}{\tau \cdot T}},
\]
(34)
where equality (b) is due to Proposition 5.1, (c) is due to Equation (18), and (d) is due to Equation (33).

Due to Equation (29), we have \(\varphi_2 > T\), since \(\lambda_2 = \frac{\Lambda - m_1 \cdot \lambda_{od}}{m - m_1} > \lambda_{od}\). Since \(\varphi_2 > 0\), we have \(x < \frac{m - \Lambda}{A \cdot \Lambda} \cdot T \cdot \tau\). Due to Equations (25) and (29), it is easy to verify \(\frac{m - \Lambda}{A \cdot \Lambda} \cdot T \cdot \tau \in (0, \tau)\). Thus, the effective range of \(x\) is as follows:
\[
x \in (0, (m - \Lambda) \cdot T \cdot \tau / (A \cdot \Lambda)).
\]
(35)

By Proposition 4.6, the prices of the first and second SLAs are
\[
p_1 = \rho = 1 \text{ and } p_2 = p_1 + (u(y, \varphi_2) - u(y, \varphi_1)) = u(y, \varphi_2) = u(1/x, \varphi_2) \overset{(a)}{=} 1 - (\varphi_2 - T) / x,
\]
(36)
where \(\varphi_1 = T\) and equality (a) is due to Equation (1). The CSP’s revenue \(G\), defined in Equation (11), is concretized as follows:
\[
G(x) = p_1 \cdot \Lambda_1 + p_2 \cdot \Lambda_2 \overset{(a)}{=} \frac{x}{\tau} \cdot \Lambda + \left(1 - \frac{1}{x} \cdot (\varphi_2 - T)\right) \cdot \frac{\tau - x}{\tau} \cdot \Lambda = \Lambda - \Lambda \cdot \frac{\tau - x}{x} \cdot (\varphi_2 - T)
\]
\[
\overset{(b)}{=} \Lambda - \frac{\Lambda}{\tau} \cdot \frac{\tau - x}{x} \cdot \left(\frac{A \cdot \Lambda}{\tau} \cdot \frac{\tau - x}{(m - \Lambda) - \frac{\Lambda \cdot A \cdot x}{\tau \cdot T}} - T\right),
\]
(37)
where equality (a) is due to Equations (28) and (36) and (b) is due to Equation (34). The CSP’s objective is to maximize Equation (37), subject to the constraint (29). The revenue \(G(x)\) and the revenue improvement \(\gamma(x)\) in Equation (30) achieve the maximum values under the same \(x = x^*\).

**Results.** The main results of this subsection are as follows. The related proofs are given in Appendix D.

**Proposition 5.4.** Let \(\psi = \sqrt{\frac{\tau}{A} \cdot \left(\frac{1}{\lambda_{od}} - \frac{1}{\lambda}\right)}\) where \(\lambda = \frac{\Lambda}{m}\), \(\lambda_{od} = \frac{T}{T + A}\) and \(A = 0.5 \cdot E[X^2]\). Given a system setting \(\{m, \Lambda, T, \tau\}\), the optimal revenue improvement is achieved when
\[
x^* = \tau \cdot (1 - \psi).
\]
(38)

Let \(\rho = \frac{\lambda}{\lambda_{od}} > 1\). The optimal revenue improvement is
\[
\gamma(x^*) = \rho - \rho \cdot \frac{T^2}{A \cdot \Lambda \cdot \tau} \cdot \frac{\rho - 1}{(1 - \psi)^2}.
\]
(39)

With Proposition 5.4, we can determine the optimal values of the other endogenous variables \(\varphi_2, p_2\) and \(\{m_t\}_{t=1}^2\) in Equations (32), (34), and (36). By Proposition 5.4 and Definition 5.5, we have

**Corollary 5.5.** The feasible operational region of a CSP is such that
\[
\tau > \frac{\rho \cdot T^2}{A \cdot \Lambda \cdot \left(1 - \sqrt{\frac{T}{A} \cdot \left(\frac{1}{\lambda_{od}} - \frac{1}{\lambda}\right)}\right)^2} = \frac{T \cdot (A + T)}{A \cdot \left(1 - \sqrt{1 + \frac{T}{A} \cdot \frac{1}{\lambda_{od}} \cdot m} \right)^2}.
\]
(40)

Corollary 5.5 shows that, only if the system setting \(\{m, \Lambda, T, \tau\}\) satisfies the relation (40), a CSP can improve its revenue by implementing a SMS-based service system. A noticeable feature is that a CSP can improve its revenue only if the overall delay-tolerance level \(\tau\) of the user population exceeds some threshold. This threshold relates to the delay \(T\) of on-demand service, the average job arrival rate per server \(\lambda = \frac{\Lambda}{m}\), and the second moment \(E[X^2]\) of the job service time distribution.
By Proposition 5.4, the larger the value of $\tau$, the higher the revenue improvement. The revenue improvement $\gamma(x^*)$ is bounded by $\rho$.

Under a given system setting $(T, m, \Lambda, \tau)$, a CSP can use Proposition 5.4 to optimally determine the proportion of jobs assigned to each SLA; then it can compute the optimal system configuration by Equations (32), (34), and (36). Also, a CSP can easily compute the optimal revenue improvement by Equation (37) to evaluate whether it can benefit from a delay-differentiated market. All in all, the resulting optimal revenue improvement $\gamma(x^*)$ is a function of the system setting, and it also has rich implications in understanding the delay-differentiated cloud market, where a system setting includes the market condition $\Lambda$ and $\tau$ and other parameters $m$ and $T$. We can see from Equation (39) that the revenue improvement depends on the average load per server $\lambda = \frac{\Lambda}{m}$, rather than any single parameter $m$ or $\Lambda$, and can also be denoted as $\gamma(T, \lambda, \tau)$. $T$ is a fixed parameter and is set to a larger value 0.02 [10].

Suppose the job service time follows an exponential distribution, i.e., $X \sim \text{Exponential}$; then we have $A = 1$ by Equation (19). The revenue improvement $\gamma$ is illustrated in Figure 6 where only the positive revenue improvement is shown. The feasible operational region of a CSP is illustrated by the blue area in Figure 7(a) and it implies that a CSP can gain under a wide range of market conditions. Given an estimated range of the market conditions, one can easily see from Figure 6 the possible revenue improvement with a delay-differentiated market. Fixing the average load per server $\lambda$, the CSP can see that the revenue improvement $\gamma$ increases as the delay-tolerance level $\tau$ increases. Given the delay-tolerance level $\tau$, the revenue improvement $\gamma$ first increases as the load $\lambda$ per server increases and then begins to decrease as the load increases further. In practice, a CSP like Amazon EC2 or Microsoft Azure often has rich capital and can adapt its capacity $m$ to maintain its load per server at a desired level to maximize the revenue [57]. Under a given delay-tolerance level $\tau$, the optimal load $\lambda$ is illustrated by the red point in Figure 7(a), and the corresponding revenue improvement is illustrated in Figure 7(b).

In addition, we consider the case where the job service time follows a log-normal distribution (i.e., $X \sim \text{Log-normal}$) and exhibits higher variability [55], which is used to examine the robustness of the SMS-based service system. We take the setting that the variance of $X$ is 5 [56] and have $A = 3$ by Equation (23). We note that, under this setting, the revenue improvement $\gamma$ brought by the PBS-based service system is upper bounded by $1 + \frac{T}{3} \approx 1.007$ by Proposition 5.2, which is small. Similarly...
Fig. 7. $X \sim \text{Exponential}$: In plot (a), the blue area illustrates the region of $\lambda$ and $\tau$ in which a CSP can achieve a revenue improvement; given the delay-tolerance level, each red point represents the optimal load per server under which a CSP achieve the maximum revenue improvement. In plot (b), given the delay-tolerance level $\tau$, each blue point illustrates the maximum revenue improvement achieved under the optimal load.

Fig. 8. Revenue improvement $\gamma$ under varying conditions ($\lambda$, $\tau$), where $X \sim \text{log-normal}$.

Fig. 9. $X \sim \text{Log-normal}$: Each plot has the same meaning as Figure 7.

to the $X \sim \text{Exponential}$ case, the revenue improvement is illustrated in Figure 8. The feasible operational region of a CSP is illustrated by the blue area in Figure 9(a); under a specific delay-tolerance level $\tau$, the maximum revenue improvement is illustrated in Figure 9(b). To sum up, the revenue improvement of the SMS-based service system in the case of $X \sim \text{Log-normal}$ is still significant under a wide range of conditions and is similar to its counterpart in the case of $X \sim \text{Exponential}$.
5.6 Optimally Configuring the SMS-based Service System

Our objective is to maximize the revenue. In this subsection, we show a procedure that can determine the optimal SLA delays and prices of a SMS-based service system when multiple SLAs are offered.

By Proposition 4.1, we have that the market is segmented by a sequence. Suppose the market segmentation is $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_{L+1}$ and the number of servers assigned to different SLAs are $m_1, m_2, \ldots, m_L$. They can uniquely determine the SLA delays and prices and the job arrival rate of each SLA. Specifically, the market segmentation determines the job arrival rate of each SLA by Equation (9). Further, for all $l \in [1, L]$, the number $m_l$ determines the job arrival rate per server of the $l$th SLA and the SLA delay $\varphi_l$ by Equation (18) and Proposition 5.1. Given the market segmentation and the SLA delays, we have shown in Section 4.2 the corresponding optimal SLA prices by Proposition 4.6. By Equation (11), the job arrival rate of each SLA and the SLA prices determine the revenue. Thus, our decision variables are $\tilde{\alpha}_2, \ldots, \tilde{\alpha}_L$ and $m_1, \ldots, m_L$ with the aim of maximizing the revenue, where $\tilde{\alpha}_1 = \overline{\alpha}, \tilde{\alpha}_{L+1} = \underline{\alpha}$, and $\sum_{l=1}^{L} m_l = m$.

The parameters $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_{L+1}$ uniquely corresponds to an element in the following set $\mathcal{A} = \{(\alpha_1, \alpha_2, \ldots, \alpha_{L+1}) | \overline{\alpha} = \alpha_1 > \alpha_2 > \cdots > \alpha_{L+1} = \underline{\alpha}, \alpha_2, \alpha_3, \ldots, \alpha_L \in \Phi\}$ where $\tilde{\alpha}_l = \alpha_l$ for all $l \in [1, L + 1]$. The parameters $m_1, m_2, \ldots, m_L$ uniquely correspond to an element in the following set $\mathcal{M} = \{(i_1, i_2, \ldots, i_{L+1}) | 0 = i_1 < i_2 < \cdots < i_{L+1} = m\}$; here, $m_l$ is set to $i_{l+1} - i_l$ for all $l \in [1, L]$. We can give a procedure to determine the optimal tuples in $\mathcal{A}$ and $\mathcal{M}$ such that the CSP achieves the maximum revenue; then, the corresponding delays and prices under these two tuples will be the optimal ones. The optimal tuples can be found by searching each possible pair of tuples in $\mathcal{A}$ and $\mathcal{M}$. The corresponding procedure is presented in Algorithm 1. Its optimality is formalized in the following proposition whose proof can be found in Appendix E.

**Proposition 5.6.** Algorithm 1 gives the optimal delays and prices of SLAs, and its time complexity is $O(m^{L-1} \cdot n^{L-1})$.

---

**Algorithm 1: Optimal Parameter Configuration**

1. $G^* \leftarrow 0$, $\mathcal{A}' \leftarrow \emptyset$, $M' \leftarrow M$; // $G^*$: record the current optimal revenue; $\mathcal{A}'$ and $M'$: record the tuples unexamined yet respectively in $\mathcal{A}$ and $M$
2. while $M' \neq \emptyset$
   3. Get a tuple $(i_1, i_2, \ldots, i_{L+1})$ from $M'$, and the $l$th group is assigned $m_l = i_{l+1} - i_l$ servers;
   4. while $\mathcal{A}' \neq \emptyset$
      5. Get a tuple $seq = (\alpha_1, \alpha_2, \ldots, \alpha_{L+1})$ from $\mathcal{A}'$;
      6. Compute the job arrival rate $\lambda_l$ of the $l$th SLA by Equation (9) and Proposition 4.1;
      7. For all $l \in [1, L]$, compute the expected job delay $\tau_l$ of the $l$th SLA by Equation (18);
      8. if $\varphi_1 \leq T < \varphi_2 < \cdots < \varphi_L$
         9. Set the delay $\varphi_l$ of the $l$th SLA to $\tau_l$ for all $l \in [2, L]$, and $\varphi_1$ to $T$;
         10. Use Proposition 4.6 to compute the optimal prices of SLAs $p_1, p_2, \ldots, p_L$;
         11. Compute the revenue $G$ by Equation (11), where $w_l = \lambda_{l-1} - \lambda_l \cdot s$;
      12. if $G > G^*$ then
         13. $G^* \leftarrow G$, $\varphi_l^* \leftarrow \varphi_l$, $p_l^* \leftarrow p_l$, $m_l \leftarrow m_l^*$, for all $l \in [1, L]$; // record the optimal SLA delays and prices, and division of servers
      14. Delete $seq$ from $\mathcal{A}'$;
      15. Delete the tuple $(i_1, i_2, \ldots, i_{L+1})$ from $M'$;
6 NUMERICAL RESULTS

In this section, we use Algorithm 1 in Section 5.6 to give some numerical results to show the performance of the proposed SMS-based service system in more cases (e.g., L ≥ 3), as a complement to the numerical results in Section 5.5 where L = 2. Specifically, in Section 6.1, we give the basic experimental setting. Related numerical results are presented in Section 6.2. Besides, we also examine the proposed service system under other possible settings; due to the page limit, please see Appendix F for the details.

6.1 Experimental Setting

There are a total of m servers. We fix the number of servers m = 100 and allocate a proper proportion of servers to each SLA. The on-demand price \( p \) (i.e., the price \( p_1 \) of the first SLA) is normalized as 1, and its delay \( T \) is 0.02. Suppose that the WTP function is a linear function in Equation (1). Given a delay–cost type \( \alpha \), let \( \varphi_z' = \frac{1}{\alpha} \) and a user’s WTP becomes zero when the delay is \( \varphi_z = \varphi_z' + T \), as described in Section 5.5; here \( \varphi_z' \) is called the relative zero-WTP point of a user of type \( \alpha \) and its value reflects the user’s delay sensitivity. There are \( n = 50 \) types of users and for all \( i \in [1, n] \) the WTP of the \( i \)th type of users becomes zero when the delay is \( \varphi_{z,i} = T + \varphi_{z,i}' \). The first type of users is the most delay-sensitive and its WTP becomes zero even if the delay is slightly larger than \( T \); thus, we set \( \varphi_{z,1}' = \epsilon \), where \( \epsilon \) is arbitrarily small. For all \( i \in [2, 50] \), we set \( \varphi_{z,i}' = (i-1)\cdot\sigma \), and have \( \varphi_{z,1} < \varphi_{z,2} < \cdots < \varphi_{z,50} \). The value of \( \sigma \) determines the delay-tolerance level of the user population, and if it is large, then the population has a high delay-tolerance level. We consider three cases where the delay-tolerance level is low, medium and high with \( \sigma = 0.02, 0.04, \) and \( 0.08 \) respectively; then, the maximum of the relative zero-WTP points is 0.98, 1.96, and 3.92, respectively.

The mean arrival rate of the jobs of all types is \( \Lambda \); the service time of a job follows an exponential distribution and their mean is normalized as one, i.e., \( s = 1 \), where \( \mathbb{E}[X^2] = 2 \) by Equation (19). Users are independently and uniformly distributed over the \( n \) types, and the mean job arrival rate of each type is \( \frac{\Lambda}{n} \). Then \( \rho = \frac{\Lambda}{m} \cdot s = \lambda \) denotes the average load per server when all \( m \) servers are considered. We denote by \( G_{\text{sms}}^* \) the optimal revenue achieved by Algorithm 1. In an on-demand service system, \( G_{\text{od}} \) denotes its revenue and is defined in Equation (26). The following ratio is the main performance metric in our experiments: \( \gamma = G_{\text{sms}}^*/G_{\text{od}} \). Specifically, if \( \gamma > 1 \), then the SMS-based service system will outperform the on-demand system; the larger the value of \( \gamma \), the higher the revenue improvement.

6.2 Numerical Results

In Section 6.2.1, we illustrate the performance of the proposed SMS-based service system in a basic case defined in Section 6.1. This setting has three features: (i) the job service time follows an exponential distribution, (ii) the users’ delay-tolerance distribution is basically uniform, and (iii) the WTP functions are linear. In each of the following three subsubsections, we will vary one feature, while keeping the other features unchanged, to check the performance of the proposed system. In this article, the mean job service time is normalized to be 1 like in References [40, 45]. In the first four subsubsections, the delay \( T \) of on-demand service is set to 0.02, i.e., this delay represents 2% of the mean job service time. In Section 6.2.5, we check the performance of the proposed system with other larger or smaller values of \( T \), which somewhat shows that the applicability of the proposed system is extensive and not limited to a specific workload.

6.2.1 Revenue Improvement in the Basic Case. In this subsubsection, we consider the basic case defined in Section 6.1. In the experiments, we vary the average load per server \( \lambda \) that increases from 0.02 with a stepsize 0.01, where 0.02 is a value slightly larger than \( \lambda_{od} \) in Equation (25). Given the number \( L \) of SLAs and the delay-tolerance level specified by \( \sigma \), we illustrate in Figure 10(a)
the maximum revenue improvement $\gamma$, which is achieved when $\lambda$ increases to some specific value; here, the red, blue, and magenta stars are for the cases of low, medium, and high delay tolerance, respectively. For example, in the low-delay-tolerance case with $L = 2$, the maximum revenue improvement is 1.829, and it is achieved when $\lambda = 0.05$. From the figure, we can see a remarkable revenue improvement ranging from 182.9% to 370.8%. In the case of same delay tolerance, we can see that (i) the larger the number $L$ of SLAs, the higher the revenue improvement $\gamma$ and (ii) the higher the delay tolerance, the higher the revenue improvement.

However, the revenue improvement when a CSP offers two SLAs (i.e., $L = 2$) is comparable to its counterparts with more SLAs offered (i.e., $L \geq 3$). This implies that, in practice, offering two SLAs may be enough to achieve a significant revenue improvement while keeping the simplicity in implementation. The service model of this article can be viewed as a complement to the on-demand service, and it can attract potential delay-tolerant users from the market and improve the revenue efficiency. We illustrate in Figure 10(b) how the revenue improvement $\gamma$ varies as the average load per server $\lambda$ increases. Observing the stars of same color, the revenue improvement $\gamma$ first increases and then begins to decrease as $\lambda$ increases.

Next, we illustrate in Figures 11(a) and 11(b) the SLA prices $p_2, \ldots, p_L$ and delays $\varphi_2, \ldots, \varphi_L$ in the low-delay-tolerance case; here the price and delay of the first SLA are 1 and 0.02. This helps understand that the users can benefit from a delay-differentiated market by trading their delay tolerance for a lower price to get services. For example, when $L = 2$, we observe the red stars and have that the price $p_2$ and delay $\varphi_2$ of the second SLA are 0.6375 and 0.09974. This means that the delay-tolerant users, who are assigned to the second SLA, can get a discount of up to 36.25%.

### 6.2.2 Revenue Improvement under Non-uniform Delay-tolerance Distributions

In the experiments of Section 6.2.1, we basically use a uniform distribution to generate the relative zero-WTP points of users, which is introduced in Section 6.1. Now we give numerical results under the other two types of distribution. The relative zero-WTP point $\varphi_{z,1}^\prime$ of the first user is still set to $\epsilon$. Then, we choose 49 values, whose range is in $[0.02, \tau]$, from a normal distribution whose mean is $x$, whose standard deviation is 2, and whose shape is symmetric about the mean $x$. We use these 49 values as the relative zero-WTP points of the other 49 users. With a given $\tau$, the two types of delay-tolerance distribution are set in the following way:

**Head intensive.** We set $x = 0.02$ and the 49 values are chosen from the right side of the mean $0.02$; then more users have smaller delay tolerances (i.e., smaller relative zero-WTP points).

**Tail intensive.** We set $x = \tau$ and the 49 values are chosen from the left side of the mean $\tau$; then more users have larger delay tolerances (i.e., larger relative zero-WTP points).
Fig. 11. Regardless of the value of $L$, the price and delay of the first SLA are 1 and 0.02. In plot (a), under the low-delay-tolerance level, the stars of same color illustrate the prices of different SLAs given the number of SLAs $L$, e.g., the red star illustrates the price of the second SLA when $L = 2$. In plot (b), the stars of same color illustrate the prices of different SLAs given the number of SLAs $L$, e.g., the red star illustrates the delay of the second SLA when $L = 2$.

Fig. 12. Both plots illustrate the case where the users’ delay-tolerance distribution is head intensive. In plots (a) and (b), the stars of same color have the same meaning as the stars in Figure 10, except that the experiments of plot (b) is taken with the medium delay-tolerance level $\tau = 1.96$.

Here, $\tau$ is an upper bound of the relative zero-WTP points of all users and represents the delay-tolerance level of the user population. In the experiments below, $\tau$ will be set to 0.98, 1.96, and 3.92, respectively, which correspond to $\sigma = 0.02, 0.04, \text{and } 0.08$ in the uniform case.

Like the experiments in Section 6.2.1, the average load per server $\lambda$ is varied; then, under each pair of $(\tau, L)$, Figure 12(a) illustrates the maximum revenue improvement in the case where the delay-tolerance distribution is head intensive, while Figure 13(a) illustrates the maximum revenue improvement in the tail-intensive case. Together with Figure 10(a), we have the following observations. First, when the delay-tolerance level is low (i.e., $\tau = 0.98$), the relative zero-WTP points of all users are densely distributed over a small interval between $\epsilon$ and 0.98; then, the distribution type has little effect on the revenue improvement, as illustrated by the red stars of Figures 10(a), 12(a), and 13(a). Second, when the delay-tolerance level is higher (i.e., $\tau = 1.96 \text{ or } 3.92$), it is observed that, under the same $(\tau, L)$, the revenue improvement $\gamma$ under the tail-intensive distribution is larger than the revenue improvement $\gamma$ under the uniform distribution, which is larger than the revenue improvement $\gamma$ under the head-intensive distribution, as illustrated by the blue or magenta stars of Figures 10(a), 12(a), and 13(a). Finally, we illustrate in Figures 12(b) and 13(b) how the revenue improvement $\gamma$ varies as the average load per server $\lambda$ increases in the head-intensive and tail-intensive cases, respectively.
6.2.3 Revenue Improvement with a Non-exponential Service Time Distribution. In this subsubsection, we consider the case where the job service time follows a log-normal distribution (i.e., $X \sim \text{Log-normal}$) and exhibits higher variability [55]. We take the setting that the variance of $X$ is 5 [56] and have $A = 3$ by Equation (23). In the experiments, we vary the average load per server $\lambda$ that increases from 0.007 with a stepsize 0.005, where 0.007 is a value slightly larger than $\lambda_{od}$ in Equation (25). Under each pair of $(\sigma, L)$, Figure 14(a) illustrates the maximum revenue improvement $\gamma$, where a significant revenue improvement is still achieved. From Figures 10(a) and 14(a), it is observed that the revenue improvements in the exponential and log-normal cases are similar. Such observation can also be obtained from Figures 7(b) and 9(b). Finally, we illustrate in Figure 14(b) how the revenue improvement $\gamma$ varies as the average load per server $\lambda$ increases.

6.2.4 Revenue Improvement under Concave WTP Functions. In this subsubsection, we give some numerical results when the concave WTP function (2) is applied. Still, like the experiments in Section 6.2.1, the average load per server $\lambda$ is varied; the related results are illustrated in Figure 15. From Figures 10(b) and 15(b), the revenue improvement under concave WTP functions exhibits similar phenomenon to its counterpart under linear WTP functions. Roughly, as the delay increases, the WTPs of users with concave WTP functions decrease more slowly than their counterparts with linear WTP functions in Section 6.2.1. Thus, from Figures 10(a) and 15(a), it is observed that, under a particular condition in terms of the number $L$ of SLAs and the population’s delay-tolerance level defined by $\sigma$, the revenue improvement under concave WTP functions is higher than its counterpart under linear WTP functions. Under the same average load per server, it is observed from Figures 10(b) and 15(b) that the revenue improvement under concave WTP functions is also higher.
Fig. 15. Both plots illustrate the case of concave WTP functions. In plots (a) and (b), the stars of same color have the same meaning as the stars in Figure 10.

Fig. 16. Both plots illustrate the revenue improvements under different values of $T$, where $\sigma$ is set to 0.04. In plots (a) and (b), the stars of same color have the same meaning as the stars in Figure 10.

The performance of the proposed SMS-based system, under the basic case where the WTP functions are linear, is a lower bound of and can serve as a guide to the performance of the proposed system under the concave WTP functions.

6.2.5 Revenue Improvement with Different $T$. In this subsubsection, we consider the effect of the delay $T$ of on-demand service on the revenue improvement. We set $T$ to three different values 0.005, 0.01, and 0.03, respectively, and take experiments with a fixed delay-tolerance level defined by $\sigma = 0.04$. In the experiments, we vary the average load per server $\lambda$ that increases from $T$ with a stepsize 0.005, where $T$ is a value slightly larger than $\lambda_{od}$ in Equation (25), since $A = 1$ by Equation (19). Given a pair of $(L, T)$, we illustrate in Figure 16(a) the maximum revenue improvement $\gamma$, which is achieved when $\lambda$ increases to some specific value. Here the numerical results with $T = 0.02$ is given by the blue stars of Figure 10(a). From Figure 16(a), it is observed that the revenue improvement is decreasing in $T$, given the value of $L$. In Reference [35], the workload traces from Google are studied where the averaged job service time can be multiple hours, e.g., 11.11 hours. With this example, $T$ can range between 0.003 and 0.0075 after normalization if the on-demand service is provided in 2–5 minutes upon request. Given the value of $L$, if the proposed system achieves a good performance when $T = 0.02$, then it achieves a better performance with $T \in [0.003, 0.0075]$, which is illustrated by the blue stars of Figure 10(a) with $T = 0.02$ and the red stars of Figure 16(a) with $T = 0.005$.

Finally, we give some observation from Figures 10(b), 12(b), 13(b), 14(b), and 15(b) where a specific value of $\sigma$ is applied in each plot and we show how the revenue improvement varies with the average job arrival rate $\lambda$ per server. It is observed that, in practice, if $T$ is set to the same value...
of 0.02, then the job arrival rate $\lambda$ per server can be set to around 0.05 with which a significant revenue improvement can be achieved; here the value of $\lambda = 0.05$ seems independent of the factors including the delay-tolerance level of users ($\tau = 0.98, 1.96, \text{ or } 3.92$), the delay-tolerance type distribution (uniform, head intensive, or tail intensive), the second moment of the job service time distribution ($E[X^2] = 2 \text{ or } 6$), and the WTP functions (linear or concave). Thus, the question on how to estimate the value of $\lambda$ to achieve a significant revenue improvement may be solved by observing these numerical results. However, it is still related to the value of the delay $T$ of on-demand services, as indicated by Figure 16(b).

7 CONCLUSION

In cloud markets, there exist both latency-critical jobs and jobs that could tolerate different degrees of delay. The resource efficiency of a system depends largely on the job’s latency requirements and the resulting resources allocation strategy. We propose a delay-differentiated pricing and service model where multiple SLAs are provided, as a complement to the existing on-demand service system. The resulting market structure is studied and we thus derive the pricing rule: under the proposed framework the mechanism is seen to be DSIC and the CSP’s revenue is maximized. We consider the PBS and SMS architectures, respectively, for fulfilling SLAs: The first appears prevalent in the literature while the second appears a rather appealing alternative due to its simplicity. We focus on non-preemptive services: The system analysis performed on these two architectures discourages the adoption of the PBS architecture and justifies the preference to the SMS-based service system. For the SMS-based system, we further leverage queueing theoretical models to determine the optimal SLA delays and prices when multiple SLA are offered. We also give a closed-form expression of the revenue improvement over the pure on-demand service model in the case that two SLAs are offered. Numerical results show that the CSP can achieve a significant revenue improvement even in the case that two SLAs are offered, which is comparable to the revenue improvement when more than two SLAs are offered.

As shown by the numerical results of Section 6.2, the revenue improvement is non-decreasing in the number of SLAs offered. One may consider how to determine the optimal number of SLAs [52, 53], which is an important question worth studying in future. It would also be interesting to extend our model to consider the case where the job service time is inversely correlated with its delay sensitivity.

APPENDICES

A PROOFS FOR MARKET PROPERTIES

A.1 Proofs Relating to Proposition 4.1

Proof of Lemma 4.2. Let $\varphi \in [T, +\infty)$. It suffices to prove the conclusion that $g(\varphi) = u(\alpha_2, \varphi) - u(\alpha_1, \varphi)$ is an increasing function of $\varphi$; then the lemma holds, since $g(\varphi_k) > g(\varphi_{k_1})$. To prove this, we note that the derivative of $g(\varphi)$ is

$$g'(\varphi) = \frac{\partial u(\alpha_2, \varphi)}{\partial \varphi} - \frac{\partial u(\alpha_1, \varphi)}{\partial \varphi}.$$  

Since $\alpha_1 > \alpha_2$, we have $g'(\varphi) > 0$ by the fourth point of Property 1, and $g(\varphi)$ is increasing.

Proof of Lemma 4.3. We prove this by contradiction. Suppose $k_2 < k_1$ and the SLA delays satisfy $\varphi_{k_2} < \varphi_{k_1}$. The user of type $\alpha_1$ (respectively, $\alpha_2$) achieves the maximum surplus under the SLA $k_1$ (respectively, $k_2$), and we thus have

$$u(\alpha_1, \varphi_{k_1}) - p_{k_1} \geq u(\alpha_1, \varphi_{k_2}) - p_{k_2}, \tag{41}$$  
$$u(\alpha_2, \varphi_{k_1}) - p_{k_1} \leq u(\alpha_2, \varphi_{k_2}) - p_{k_2}. \tag{42}$$
Multiplying Equation (41) by \(-1\) and adding the resulting inequality to Equation (42), we have \(u(\alpha, \varphi_{k_1}) - u(\alpha, \varphi_{k_2}) \leq u(\alpha, \varphi_{k_2}) - u(\alpha, \varphi_{k_1})\). However, since \(\alpha_1 > \alpha_2\) and \(k_2 < k_1\), we have by Lemma 4.2 that \(u(\alpha, \varphi_{k_1}) - u(\alpha, \varphi_{k_2}) > u(\alpha, \varphi_{k_2}) - u(\alpha, \varphi_{k_1})\), which contradicts the previous inequality.

**Proof of Proposition 4.1.** Each type of users will be assigned to some SLA, and \(\Phi_l\) denotes the set of the types of the users assigned to the \(l\)th SLA for all \(l \in [1, L]\). Let \(\hat{\alpha}_l\) denote the maximum type in \(\Phi_l\) such that only the users of type \(\alpha \leq \hat{\alpha}_l\) will possibly be assigned to the \(l\)th SLA. For all \(l \in [1, L - 1]\), when the users of types \(\hat{\alpha}_l\) and \(\hat{\alpha}_{l+1}\) are respectively assigned the \(l\)th and \((l + 1)\)-th SLAs, we have by Lemma 4.3 that \(\hat{\alpha}_l > \hat{\alpha}_{l+1}\), which can be easily proved by contradiction. A user of type \(\overline{\alpha}\) will be assigned to a SLA whose number is no larger than one (i.e., the first SLA), since \(\overline{\alpha} \geq \hat{\alpha}_1\). Thus, we have \(\hat{\alpha}_1 = \overline{\alpha}\).

By Lemma 4.3, we also have that (i) for all \(l \in [1, L - 1]\) every user of type \(\alpha \in (\hat{\alpha}_{l+1}, \hat{\alpha}_l] \cap \Phi\) will be assigned to a SLA whose number \(l'\) is no smaller than \(l\) but no larger than \(l + 1\), and (ii) every user of type \(\alpha \in (\overline{\alpha}, \hat{\alpha}_L] \cap \Phi\) will be assigned to a SLA whose number is no smaller than \(L\), since \(\alpha \leq \hat{\alpha}_L\). In the first case, \(\alpha > \hat{\alpha}_{l+1}\) and \(\hat{\alpha}_{l+1}\) is the maximum type of \(\Phi_{l+1}\); thus \(l'\) will be smaller than \(l + 1\) and equal \(l\). The proposition thus holds.

### A.2 Proofs Relating to Proposition 4.5 and 4.6

**Proof of Proposition 4.5.** In the case that \(\alpha \neq \hat{\alpha}_l\), we have by Lemma 4.7 the conclusion that, (i) for all \(l' \in [2, L]\), the user achieves a higher surplus under \(l'\)th SLA than under the \((l' - 1)\)-th SLA and (ii) for all \(l' \in [L, L - 1]\), it achieves a higher surplus under the \(l'\)th SLA than under the \((l' + 1)\)th SLA; thus, the user achieves the highest surplus under the \(l\)th SLA. In the case that \(\alpha = \hat{\alpha}_l\), we still have the above conclusion, except that the user achieves the same surplus under the \(l\)th and \((l - 1)\)-th SLAs when \(l' \in [2, L]\) and \(l' = l\); thus, the user achieves the maximum surplus under both the \(l\)th and \((l - 1)\)-th SLAs. According to Definition 3.1, the proposition holds in both cases.

**Proof of Proposition 4.6.** Let us consider a user of type \(\alpha \in \Phi_l\) who reports to the CSP that its type is \(\alpha'\). No matter what the other users do, we have by Proposition 4.5 that it achieves the maximum surplus under the \(l\)th SLA and will be assigned by the CSP to the \(l\)th SLA when it truthfully reports its type, i.e., \(\alpha' = \alpha\). Thus, it cannot gain more by misreporting its type, since misreport can lead to that it is assigned to the \(l\)th SLA or the other SLAs. The first point thus holds by Definition 3.2.

The objective of our framework is to maximize Equation (11); given the market segmentation \(\hat{\alpha}_1, \ldots, \hat{\alpha}_{l+1}\) defined in Proposition 4.1, the job arrival rate of each SLA is fixed by Equation (9), and we have the conclusion that the larger the SLA prices, the larger the value of \(G\). The first SLA’s price \(p_1\) is fixed and equals \(p\). To guarantee the truthfulness of the users of type \(\alpha \in \Phi_l\), a necessary condition is that

\[
u(\alpha, \varphi_{l-1}) - p_{l-1} \leq u(\alpha, \varphi_1) - p_1, \text{ for all } l \in [2, L].
\]

Then, it holds for all \(\alpha \in \Phi_l\) that

\[
p_l \leq p_{l-1} - (u(\alpha, \varphi_{l-1}) - u(\alpha, \varphi_1));
\]

by Lemma 4.2, \(u(\alpha, \varphi_{l-1}) - u(\alpha, \varphi_1)\) achieves the maximum value when \(\alpha = \hat{\alpha}_l\). Further, irrespective of the value of \(p_{l-1}\), the maximum possible value of \(p_1\) is \(\hat{p}_1\) for all \(l \in [2, L]\). Thus, the second point holds.

**Proof of Lemma 4.7.** In the first case, if \(\alpha = \hat{\alpha}_l\) and \(l' = l\), then the surplus difference of the user under the \(l'\)th and \((l' - 1)\)-th SLAs is \((u(\hat{\alpha}_l, \varphi_1) - p_l) - (u(\hat{\alpha}_l, \varphi_{l-1}) - p_{l-1})\); it equals zero due to Definition 4.4. Otherwise, we have either \(\alpha < \hat{\alpha}_l\) or \(l' < l\): In the former, \(\alpha < \hat{\alpha}_l \leq \hat{\alpha}_{l'}\), since \(l' \in [2, l]\); in the latter, \(\alpha \leq \hat{\alpha}_l < \hat{\alpha}_{l'}\). Thus, we have \(\alpha < \hat{\alpha}_{l'}\). The user’s surplus difference under
two adjacent SLAs $l'$ and $l' - 1$ is

$$
(u(\alpha, \varphi_{l'}) - p_{l'}) - (u(\alpha, \varphi_{l'-1}) - p_{l'-1})
$$

$$
\leq (u(\tilde{\alpha}_{l'}, \varphi_{l'-1}) - u(\tilde{\alpha}_{l'}, \varphi_{l'})) - (u(\alpha, \varphi_{l'-1}) - u(\alpha, \varphi_{l'})) \quad (b)
$$

here equation (a) is due to Definition 4.4 and (b) is due to Lemma 4.2. In the second case, we have

$$
\tilde{\alpha}_{l'+1} < \alpha, \text{ since } \alpha \in (\tilde{\alpha}_{l+1}, \tilde{\alpha}_l) \text{ and } l' \geq l, \text{ and the user's surplus difference under the } (l' + 1)\text{-th SLAs is}
$$

$$
(u(\alpha, \varphi_{l'}) - p_{l'}) - (u(\alpha, \varphi_{l'+1}) - p_{l'+1})
$$

$$
\leq (u(\alpha, \varphi_{l'}) - u(\alpha, \varphi_{l'+1})) - (u(\tilde{\alpha}_{l'+1}, \varphi_{l'}) - u(\tilde{\alpha}_{l'+1}, \varphi_{l'+1})) \quad (d)
$$

here equation (c) is due to Definition 4.4 and (d) is due to Lemma 4.2. Hence, the lemma holds.

B PROOF OF PROPOSITION 5.1

We prove this by contradiction. We have $\varphi_{l'} \geq t_l$ for all $l \in [2, L]$. Let us consider an optimal solution where the SLA delays and prices are $\varphi_{l'}^*$ and $p_{l'}^*$ for all $l \in [2, L]$, and the market segmentation is $\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_{l'}$. Suppose there exists some SLA $l \in [2, L]$ such that $\varphi_{l'} > t_l$; let $l'$ denote the minimum such $l$, where $\varphi_{l'}^* = t_2, \ldots, \varphi_{l'-1}^* = t_{l'-1}$ if $l' > 2$. If we decrease the delay of the $l'$th SLA to $t_l'$ and keep the others unchanged, then we denote the corresponding prices by $\tilde{p}_{l'}, \ldots, \tilde{p}_{L}$. It suffices to prove the conclusion that $\tilde{p}_{l'} > p_{l'}^*$ for all $l \in [l', L]$ and $\tilde{p}_{l'} = p_{l'}^*$ for all $l \in [2, l'-1]$, if $l' > 2$. This will lead to that the revenue (Equation (11)) increases, which contradicts the assumption that $p_{l'}^*, \ldots, p_{L}^*$ are optimal; the proposition thus holds. Now we prove the conclusion. The SLA prices are determined by Proposition 4.6. First, we have $p_{l'}^* = \tilde{p}_{l'}$ for all $l \in [2, l'-1]$ if $l' > 2$; this is due to that $\varphi_{l'}^*, \ldots, \varphi_{l'-1}^*$ does not change. Second, for the $l'$th SLA, we have

$$
\tilde{p}_{l'} = \tilde{p}_{l'-1} + u(\tilde{\alpha}_{l'}, t_{l'}) - u(\tilde{\alpha}_{l'}, t_{l'-1}) \quad (a)
$$

The inequality (a) is due to that $\tilde{p}_{l'-1} = p_{l'-1}^*, u(\tilde{\alpha}_{l'}, t_{l'}) > u(\tilde{\alpha}_{l'}, \varphi_{l'}^*)$, and $t_{l'-1} = \varphi_{l'-1}^*$. Third, for the $(l'+1)$-th SLA, we have

$$
\tilde{p}_{l'+1} = \tilde{p}_{l'} + u(\tilde{\alpha}_{l'+1}, \varphi_{l'}^*) - u(\tilde{\alpha}_{l'+1}, t_{l'})
$$

$$
= \tilde{p}_{l'-1} + u(\tilde{\alpha}_{l'}, t_{l'}) - u(\tilde{\alpha}_{l'}, \varphi_{l'}^*) + u(\tilde{\alpha}_{l'+1}, \varphi_{l'}^*) - u(\tilde{\alpha}_{l'+1}, t_{l'}) \quad (b)
$$

$$
\geq p_{l'+1}^* + u(\tilde{\alpha}_{l'}, \varphi_{l'}^*) - u(\tilde{\alpha}_{l'}, \varphi_{l'}^*) + u(\tilde{\alpha}_{l'+1}, \varphi_{l'}^*) - u(\tilde{\alpha}_{l'+1}, \varphi_{l'}^*) = p_{l'+1}^*.
$$

Here the inequality (b) is due to Lemma 4.2. Fourth, if $l' + 2 \leq L$, for all $l \in [l' + 2, L]$, then we have by a simple mathematical induction that

$$
\tilde{p}_l = \tilde{p}_{l-1} + u(\tilde{\alpha}_l, \varphi_{l-1}) - u(\tilde{\alpha}_l, \varphi_{l-1}^*) \quad (c)
$$

$$
\geq p_{l-1}^* + u(\tilde{\alpha}_l, \varphi_{l-1}^*) - u(\tilde{\alpha}_l, \varphi_{l-1}^*) = p_l^*.
$$

Here the inequality (c) is due to $\tilde{p}_{l-1} > p_{l-1}^*$.

C THE PERFORMANCE OF A PBS-BASED SERVICE SYSTEM

C.1 Proof of Proposition 5.2

In a PBS-based service system, all jobs of different SLAs are commonly executed on the $m$ servers. In this article, the mean job service time $s$ is normalized to be 1. The first SLA offers service at a fixed price $p$ and guarantees a small delay of at most $T$, and we have by Equation (16) that $\varphi_1 = \lambda \cdot A/(1 - \lambda_1) \leq T$, where $0 < \lambda_1 < \lambda < 1$. Thus, we get

$$
\lambda < T/A.
$$

(43)
A CSP’s revenue $G_{pbs}$ is given in Equation (11) and we can get an upper bound for $G_{pbs}$:

$$G_{pbs} = \sum_{l=1}^{L} p_l \cdot m \cdot \lambda_l \leq p \cdot m \cdot \sum_{l=1}^{L} \lambda_l$$

where equality (a) is due to Equation (14); (b) is due to Equation (4), i.e., $p_l \leq p$ for all $l \in [1, L]$; (c) is due to Equations (6), (8), (14), and (15); and (d) is due to Equation (43). By Equations (26), (27), and (44), we have

$$\gamma = \frac{G_{pbs}}{G_{od}} = \frac{A + T}{m \cdot p \cdot T} < 1 + \frac{T}{A}.$$  (45)

Proposition 5.2 thus holds.

C.2 Illustration

We vary the value of $\pi_1$ from 0.2 to 0.95 with a step size 0.05 and compute the corresponding value of $\pi_2$ by Equation (20); then we can get the value of $A$ by Equation (21), which is illustrated by the red stars in Figure 17, where $A > 1$.

![Fig. 17. The value of $A$ under varying $\pi_1$.](image)

D PROOFS FOR THE CLOSED-FORM RESULT

We first give some preliminaries to prove Proposition 5.4. By Equations (30) and (37), we have

$$\gamma(x) = \frac{A + T}{m \cdot T} \cdot \frac{A}{\Lambda} - \frac{A + T}{m \cdot T} \cdot \frac{\Lambda}{\tau} \cdot \frac{\tau - x}{x} \cdot \left( \frac{A \cdot \Lambda}{\tau} \cdot \frac{\tau - x}{m - \Lambda} \cdot \frac{\Lambda \cdot \Lambda}{\tau \cdot \tau} \cdot x - T \right).$$  (46)

We will derive the derivative of $\gamma(x)$, denoted by $\frac{\partial \gamma}{\partial x}$ (see Section D.1). Fortunately, the roots of $\frac{\partial \gamma}{\partial x} = 0$ satisfy some nice properties (see Section D.2). We can thus derive the monotonicity of $\gamma(x)$ and obtain when $\gamma(x)$ achieves the maximum value (see the proof of Proposition 5.4 below). The proof of Corollary 5.5 follows from Proposition 5.4 and Definition 5.3.

D.1 The Derivative of the Revenue Improvement Function

**Lemma D.1.** *The derivative of $\gamma(x)$ is as follows:*

$$\frac{\partial \gamma}{\partial x} = \frac{(A + T) \cdot \Lambda^2}{\tau \cdot T} \cdot \frac{a \cdot x^2 + b \cdot x + c}{x^2 \cdot (m - \Lambda - \frac{\Lambda \cdot \Lambda}{\tau \cdot \tau} \cdot x)^2},$$  (47)

where

$$a = \frac{A^2 \cdot \Lambda}{m \cdot \tau - \frac{A}{\tau} + \frac{A \cdot \Lambda}{m \cdot \tau}}, \quad b = 2 \cdot A \cdot \left( 1 - \frac{\Lambda}{m} - \frac{A \cdot \Lambda}{m \cdot T} \right), \quad c = \left( A + T - \frac{T \cdot m}{\Lambda} \right) \cdot \left( 1 - \frac{\Lambda}{m} \right).$$

ACM Trans. Model. Perform. Eval. Comput. Syst., Vol. 8, No. 3, Article 6. Publication date: June 2023.
Delay and Price Differentiation in Cloud Computing

\[ B = \frac{(A + T) \cdot \Lambda}{T \cdot m \cdot \tau}, \quad C = \frac{A \cdot \Lambda}{T \cdot m \cdot \tau}, \quad D = m - \Lambda, \quad \text{and} \quad E = \frac{A \cdot \Lambda}{T \cdot \tau}. \quad (48) \]

\[ \gamma(x) = \left( \frac{\tau - x}{x} \cdot \left( \frac{\tau - x}{D - E \cdot x} - T \right) \right). \quad (49) \]

Then \( \frac{\partial \gamma}{\partial x} \) and \( \frac{\partial \gamma}{\partial x} \) have the following relation:

\[ \frac{\partial \gamma}{\partial x} = B \cdot \frac{\partial \gamma}{\partial x}. \]

First, some components of \( \frac{\partial \gamma}{\partial x} \) are listed below:

\[ \gamma_1(x) = C \cdot \tau \cdot \frac{\tau - x}{x \cdot (D - E \cdot x)} \],
\[ \gamma_2(x) = C \cdot \frac{\tau - x}{D - E \cdot x}, \quad \text{and} \quad \gamma_3(x) = \frac{T \cdot \tau}{x}. \]

Then we have

\[ \gamma(x) = \gamma_1(x) - \gamma_2(x) - \gamma_3(x) + T. \quad (50) \]

Second, we give the derivatives of these components:

\[ \frac{\partial \gamma_1}{\partial x} = C \cdot \tau \cdot \frac{-x \cdot (D - E \cdot x) - (\tau - x) \cdot (D - 2 \cdot E \cdot x)}{x^2 \cdot (D - E \cdot x)^2} = \frac{C \cdot \tau \cdot (-E \cdot x^2 + 2 \cdot E \cdot \tau \cdot x - \tau \cdot D)}{x^2 \cdot (D - E \cdot x)^2}, \quad (51a) \]

\[ \frac{\partial \gamma_2}{\partial x} = \frac{-(D - E \cdot x) + E \cdot (\tau - x)}{(D - E \cdot x)^2} = C \cdot \frac{\tau \cdot E - D}{(D - E \cdot x)^2} \cdot x^2 \quad (51b) \]

\[ \frac{\partial \gamma_3}{\partial x} = \frac{T \cdot \tau}{x^2} = -\frac{T \cdot \tau \cdot (D^2 - 2 \cdot D \cdot E \cdot x + E^2 \cdot x^2)}{x^2 \cdot (D - E \cdot x)^2}. \quad (51c) \]

Third, by Equation (50), we get the derivative of \( \frac{\partial \gamma}{\partial x} \):

\[ \frac{\partial \gamma}{\partial x} = B \cdot \frac{\partial \gamma}{\partial x} = B \cdot \left( \frac{\partial \gamma_1}{\partial x} - \frac{\partial \gamma_2}{\partial x} - \frac{\partial \gamma_3}{\partial x} \right) \]
\[ = B \cdot \frac{(DC + E^2 T \tau - 2EC \tau) x^2 + 2E(CT^2 - DT \tau) x + (TD^2 \tau - DC^2 \tau^2)}{x^2 (D - Ex)^2} \]
\[ = B \cdot \frac{\hat{a} \cdot x^2 + \hat{b} \cdot x + \hat{c}}{x^2 \cdot (D - E \cdot x)^2}. \]

where we have by Equation (48) that

\[ \hat{a} = \frac{\Lambda^2 \cdot A^2}{T \cdot \tau} + (m - \Lambda) \cdot \frac{A \cdot \Lambda}{T \cdot \tau}, \quad \hat{b} = 2 \cdot \left( \frac{A^2 \cdot \Lambda^2}{T} - (m - \Lambda) \cdot A \cdot \Lambda \right), \quad \hat{c} = \tau \cdot (m - \Lambda) \cdot (T \cdot m - T \cdot \Lambda - A \cdot \Lambda). \]

We can get Equation (47) from Equation (52). □
D.2 Properties of Roots

Let \( y'(x) = a \cdot x^2 + b \cdot x + c \), which is a part of Equation (47). By Equation (35), the effective range of \( x \) is \((0, \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau)\) in which \( \frac{\partial y}{\partial x} \) has the same sign as \( y'(x) \). In the following, we will study the quadratic equation \( y'(x) = 0 \), in terms of its quadratic coefficient, determinant, and roots. We can thus derive the sign of \( y'(x) \) and the monotonicity of \( y(x) \) in \((0, \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau)\). Let

\[
\lambda = \frac{\Lambda}{m} \quad \text{and} \quad \rho = \frac{\lambda}{\lambda_{od}},
\]

where \( \lambda_{od} = \frac{T}{A \cdot T} \), which enable us to simplify the expressions of \( a, b, \) and \( c \):

\[
a = \frac{A}{\tau} \cdot (\rho - 1), \quad b = 2 \cdot A \cdot (1 - \rho), \quad c = T \cdot \tau \cdot (\rho - 1) \cdot \left( \frac{1}{\lambda} - 1 \right).
\]

The discriminant of the quadratic equation \( y'(x) = 0 \) is

\[
\Delta = b^2 - 4 \cdot a \cdot c = 4 \cdot A^2 \cdot (\rho - 1)^2 \cdot \frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right).
\]

If \( \Delta \geq 0 \), then its two roots are denoted by \( x_1 \) and \( x_2 \):

\[
x_1 = \frac{-b - \sqrt{\Delta}}{2 \cdot a} = \tau \cdot \left( 1 - \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \right) \quad \text{and} \quad x_2 = \frac{-b + \sqrt{\Delta}}{2 \cdot a} = \tau \cdot \left( 1 + \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \right),
\]

where \( x_1 < x_2 \).

**Lemma D.2.** Regarding the quadratic equation \( y'(x) = 0 \), we have that

- its quadratic coefficient and discriminant are positive, i.e., \( a > 0 \) and \( \Delta > 0 \).
- its two roots satisfy \( x_1 \in (0, \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau) \) and \( x_2 \in (\tau, +\infty) \).

**Proof.** By Equations (29) and (53), we have \( \lambda > \lambda_{od} \) and \( \rho > 1 \). By Equations (54) and (55), we have \( a > 0 \) and \( \Delta > 0 \). We have

\[
0 < \frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right) \overset{(b)}{=} \sqrt{1 - \frac{T}{A} \cdot \left( \frac{1}{\lambda} - 1 \right)} \overset{(c)}{<} 1.
\]

The relation (a) is due to \( 0 < \lambda_{od} < \lambda \), (c) is due to \( \lambda \in (0, 1) \), and (b) is due to \( \lambda_{od} = \frac{T}{A \cdot T} \). Thus, we have \( x_1 \in (0, \tau) \) and \( x_2 \in (\tau, +\infty) \).

Now we prove \( x_1 < \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau \) as follows:

\[
\frac{m - \Lambda}{A \cdot \Lambda} \cdot T \cdot \tau - x_1 = \frac{m - \Lambda}{A \cdot \Lambda} \cdot T \cdot \tau - \tau \cdot \left( 1 - \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \right)
\]

\[
= \frac{\tau \cdot T}{A} \cdot \left( \frac{m - \Lambda}{A \cdot \Lambda} - \frac{A + T}{T} + \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \right)
\]

\[
\overset{(d)}{=} \frac{\tau \cdot T}{A} \cdot \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \cdot \left( 1 - \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \right) \overset{(e)}{>} 0.
\]

The relation (d) is due to Equations (25) and (53); (e) is due to Equation (57). \( \square \)
D.3 Proofs Relating to Proposition 5.4 and Corollary 5.5

Proof of Proposition 5.4. By Lemma D.2, we have that $y'(x) > 0$ when $x \in (0, x_1)$ and $y'(x) < 0$ when $x \in (x_1, \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau)$. $\frac{\partial y}{\partial x}$ has the same sign as $y'(x)$. Thus, we have $y(x)$ is increasing in $(0, x_1)$ and decreasing in $(x_1, \frac{m-\Lambda}{A \cdot \Lambda} \cdot T \cdot \tau)$, and it achieves the maximum value when $x = x_1$, i.e., $x^* = x_1$. We also use the parameters $\rho$ and $\lambda$ in Equation (53) to simplify the expression of $y(x)$ in Equation (46) and have that

$$y(x) = \rho - \rho \cdot \frac{x - \tau}{x} \cdot \frac{1}{\tau} \left( \frac{A \cdot \Lambda - \frac{A \cdot \Lambda \cdot x}{\tau}}{(m - \Lambda) - \frac{A \cdot \Lambda}{\tau} \cdot x} - T \right)$$

$$= \rho - \rho \cdot \frac{x - \tau}{x} \cdot \frac{1}{\tau} \left( A \cdot \Lambda - (m - \Lambda) \cdot T \right)$$

$$= \rho - \rho \cdot \frac{x - \tau}{x} \cdot \frac{T}{\lambda} \cdot \left( \frac{1}{\lambda} - 1 \right) \cdot \tau - \frac{T}{\lambda} \cdot x.$$  

Let $\psi = \sqrt{A \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\lambda} \right)} \in (0, 1)$ by Equation (57); here we have $\frac{T}{\lambda} \cdot \psi^2 = \frac{1}{\lambda_{od}} - \frac{1}{\lambda}$. With $x^* = x_1 = \tau \cdot (1 - \psi)$, the maximum revenue improvement is

$$y(x^*) = \rho - \rho \cdot \psi \cdot \frac{T}{1 - \psi} \cdot \frac{1}{\lambda} \cdot \left( \frac{1}{\lambda} - 1 \right) \cdot \tau - \frac{T}{\lambda} \cdot (1 - \psi) \cdot \tau$$

$$= \rho - \rho \cdot \psi \cdot \frac{T}{1 - \psi} \cdot \frac{1}{\lambda} \cdot \left( \frac{1}{\lambda} - 1 \right) \cdot \tau$$

$$= \rho - \rho \cdot \frac{T}{A \cdot \lambda} \cdot \frac{1}{\lambda_{od}} \cdot \left( \frac{1}{\lambda} - 1 \right) \cdot \tau \cdot (1 - \psi)^2.$$  

Proof of Corollary 5.5. In the feasible operational region, a CSP can improve its revenue with the SMS-based service system, i.e., $y(x^*) > 1$, and we have

$$\rho - \rho \cdot \frac{T^2}{A \cdot \lambda} \cdot \frac{1}{\lambda_{od}} \cdot (1 - \psi)^2 > 1.$$  

Thus, the feasible operational region is as follows:

$$\tau > \frac{\rho \cdot T^2}{A \cdot \lambda \cdot \left( 1 - \sqrt{\frac{T}{A} \cdot \left( \frac{1}{\lambda_{od}} - \frac{1}{\tau} \right) \cdot \left( \rho - \frac{1}{(1 - \psi)^2} \right) } \right)^2} = \frac{T \cdot (A + T)}{A \cdot \left( 1 - \sqrt{\frac{T}{A} \cdot \frac{T}{A} \cdot \left( \rho - \frac{1}{(1 - \psi)^2} \right) } \right)^2}.$$  

E PROOF OF PROPOSITION 5.6

Algorithm 1 searches each possible pair of $(a_1, a_2, \ldots, a_{L+1})$ and $(i_1, i_2, \ldots, i_{L+1})$, respectively, in $A$ and $M$ (lines 1, 2, 3, 15, 4, 5, and 14 of Algorithm 1), and computes the corresponding revenue under this pair (lines 6–11). Among all pairs that have been searched so far, it records the current maximum revenue and the corresponding SLA delays and prices and the numbers of servers assigned to SLAs (lines 1, 12, and 13). Thus, the algorithm will return the optimal solution. The sizes of $M$ and $A$ are respectively polynomial in $m$ and $n$ (i.e., $m_{L-1}$ and $n_{L-1}$). The loop in line 4 is nested in the loop in line 2; hence, the time complexity is $O(m^{L-1} \cdot n^{L-1})$.

F MORE EXPERIMENTS

In this section, we examine the proposed service system under other possible settings. In Section F.1, we show the performance of our service model under the architecture of References [9–11],
which is a hybrid of the PBS and SMS architectures. Our service system is built on some task assignment policies used in practice, as shown in Section 5.1.1. In Section F.2, we show the performance of our service model under other advanced policies that may be of future interest.

F.1 Comparison with a Hybrid Architecture

As seen in Section 1, our framework differs from References [9–11] in several aspects. Nevertheless, the service model in Section 3 and 4 is generic. The architecture of References [9, 10] can be adapted to our model, and roughly viewed as a hybrid of the PBS and SMS architectures. Specifically, all servers are separated into two parts: The first are used to fulfill the first SLA, as done by the first group of the SMS architecture; the second use priority queues to fulfill the SLAs 2, . . . , L, as done by the PBS architecture. Specially, when the number of SLAs is 2 (i.e., L = 2), the SMS and hybrid architectures are the same and the model has the same performance under both architectures, which can achieve a significantly larger revenue than the pure on-demand service model. Generally, the PBS-based service system performs worse than the SMS-based system; it can be expected that the hybrid architecture has an in-between performance, as shown later.

We still consider the case where the WTP function is Equation (1) and denote by \( G^*_{hyb} \) the maximum revenue achieved by our service model under the hybrid architecture. For all \( l \in [2, L] \), let \( \hat{\lambda}'_l \) denote the total job arrival rate of SLAs 2, . . . , l at a single server; we can derive the expected delay \( t_l \) of the lth SLA by Equations (16) and (19) and have

\[
\hat{\gamma} \coloneqq \frac{G^*_{hyb}}{G^*_{sms}}.
\]

If \( \hat{\gamma} \leq 1 \), then the service model under the hybrid architecture will be no better than the SMS-based service system. This is exactly shown by our numerical results illustrated in Figure 18.

The reason for \( \hat{\gamma} \leq 1 \) is mainly due to the correlation of the SLA delays in the hybrid architecture, which leads to that all SLAs have larger delays and thus smaller prices. We have by Equation (62) that the expected delays \( t_2, \ldots, t_L \) are all constrained by the total job arrival rate \( \hat{\lambda}'_L \), which is the average load per server in the second part. For example, in the low delay-tolerance case with \( L = 4 \), the first and second parts have 51 and 49 servers, respectively. The market segmentation
is \((\hat{a}_2, \hat{a}_3, \hat{a}_4) = (0.122, 0.36, 0.66)\). For the aggregate job arrival rates of the second part, we have \((\hat{\lambda}_2', \hat{\lambda}_3', \hat{\lambda}_4') = (0.01429, 0.03878, 0.08163)\). Due to the effect of \(\hat{\lambda}_4'\) on the delay, the second, third, and fourth SLAs have similar yet large delays and we have \((\varphi_2, \varphi_3, \varphi_4) = (0.08282, 0.08616, 0.09248)\). Such SLA delays further lead to lower SLA prices \((p_2, p_3, p_4) = (0.6859, 0.6761, 0.6652)\) and a lower revenue improvement than the SMS-based service system.

In contrast, the delays of different SLAs in the SMS architecture are independent by Equation (18). The numbers of servers assigned to different SLAs are \((m_1, m_2, m_3, m_4) = (47, 15, 18, 20)\). The market segmentation is \((\hat{a}_2, \hat{a}_3, \hat{a}_4) = (0.1, 0.32, 0.56)\). The average job arrival rates per server of the second, third, and fourth SLAs are \((\lambda_2, \lambda_3, \lambda_4) = (0.04000, 0.06667, 0.1150)\). By Reference (18), we have that the SLA delays of the second and third SLAs are smaller: \((\varphi_2, \varphi_3, \varphi_4) = (0.04167, 0.07143, 0.1299)\). This leads to that the prices of the second and third SLAs are larger, achieving a higher revenue improvement: \((p_2, p_3, p_4) = (0.8796, 0.7804, 0.6721)\). The revenue ratio is \(\bar{\gamma} = 0.8624\).

### F.2 Advanced Task Assignment Policies

The Random and Round-Robin policies are two typical policies adopted in real cloud service systems, as introduced in Section 5.1.1. Alternatively, there may also be other advanced polices worth exploring. The choice of polices needs to find a right tradeoff between the actual implementing cost, the scalability and the performance, which may be of independent interest in future. In this subsection, we take the Power of Two Choices (PTC) policy as an example to illustrate the performance of our service system under such advanced policies. The PTC policy works as follows: For every arriving job \(j\), randomly choose two of the \(m\) servers, probe them, and assign it to the server with less queued jobs. This policy is interesting in that it has been implemented in modern cluster systems and performance engineering tools [43, 44], although it brings about additional communication and coordination overheads and we have not seen its application to public cloud service systems.

For the PTC policy, analytical results are available when the job arrival follows a poisson process and the job service time follows an exponential distribution with mean \(s = 1\) [45]. Suppose under the SMS architecture that the mean job arrival rate per server of the \(l\)th SLA is \(\lambda_l\) where \(l \in [1, L]\). We have that the expected delay of the \(l\)th SLA is

\[
T_l = \sum_{i=2}^{\infty} \lambda_l^{2i-2} = \lambda_l^2 + \lambda_l^6 + \lambda_l^{14} + \cdots .
\]  
(63)

When the PTC policy is applied to the SMS-based service system, we can still use Algorithm 1 to determine the optimal system configuration, with Equation (18) replaced by Equation (63); here we still use the linear functions (1) as the WTP functions. \(\lambda_{od}\) denotes the job arrival rate per server of the on-demand market. Since the maximum possible delay of the on-demand market is \(T\), we have that \(\lambda_{od}^2 \leq T\) and an upper bound of \(\lambda_{od}\) is \(\sqrt{T}\). Thus, the revenue of an on-demand market \(G_{od}\) is bounded and approximated by \(m \cdot \sqrt{T}\). \(G\) denotes the revenue of a SMS-based service system. \(\frac{G}{m \cdot \sqrt{T}}\) is a lower bound of \(\bar{G}_{od}\) and will be used to indicate the revenue improvement \(\gamma\) in this subsection.

Similarly to Section 6.2.1 and 6.2.4, we vary the average load per server \(\lambda\) that increases from 0.15 with a step size 0.01 and calculate the revenue improvement \(\gamma\). The related results are illustrated in Figure 19; specifically, given the number of SLAs \(L\) and the population’s delay-tolerance level, Figure 19(a) illustrates the maximum revenue improvement \(\gamma\) achieved while varying the value of \(\lambda\). From Figure 19, a delay-differentiated market under the PTC policy exhibits similar patterns to its counterpart under the random or Round-Robin policy and can still achieve a significantly higher revenue improvement.
Fig. 19. Both plots illustrate the case when the PTC policy is used. In plots (a) and (b), the stars of same color have the same meaning as the stars in Figure 10.

Fig. 20. Both plots illustrate the results in the low delay-tolerance case. In plot (a), the stars illustrate the improvement \( \nu \) to the utilization of servers respectively with \( L = 2, 3, 4, \) and 5, when the Random and PTC policy are applied to the SMS-based service system. In plot (b), given the number of SLAs \( L \), the stars illustrate the average price \( \hat{p} \).

We can also see from Figures 19(a) and 10(a) that, under the same condition in terms of the number of SLAs and the population’s delay tolerance, the revenue improvement under the PTC policy is lower. The reason is as follows. The PTC policy can achieve an exponential improvement to the waiting time that a job spends in a queueing system. This implies that an on-demand service system under the PTC policy can achieve a higher utilization of servers, given the delay \( T \) that it guarantees. Thus, such advanced policies are good for an on-demand service system, in terms of the utilization and revenue. However, we define the utilization improvement \( \nu \) as the ratio of the average load per server \( \lambda \) under the SMS-based system to the average load per server \( \lambda_{od} \) under the on-demand system, i.e., \( \nu = \lambda / \lambda_{od} \), where \( s = 1 \). The average price \( \hat{p} \) of utilizing a server per unit of time is \( G_{sms}^*/(m \cdot \lambda) \). The related results in the low delay-tolerance case are illustrated in Figure 20; here the random and Round polices have the same performance. By delay differentiation, our SMS-based service system under the Random or Round-Robin policy can achieve a higher utilization improvement than the system under the PTC policy. Thus, given the number of SLAs \( L \), in the case that the average value \( \hat{p} \) that is obtained from every server is similar, the SMS-based service system under the Random or Round-Robin policy achieves a higher revenue improvement.

REFERENCES

[1] Gartner Says Worldwide IaaS Public Cloud Services Market Grew 41.4% in 2021. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2022-06-02-gartner-says-worldwide-iaas-public-cloud-services-market-grew-41-percent-in-2021.

[2] R. Buyya, S. N. Srirama, G. Casale, and R. Calheiros, et al. 2019. A manifesto for future generation cloud computing: Research directions for the next decade. ACM Comput. Surv. 51, 5, Article 105 (January 2019), 38 pages.
[3] D. Bertsekas and R. Gallager. 1987. In Data Networks. Prentice-Hall, Upper Saddle River, NJ.

[4] R. Zhou, Z. Li, C. Wu, and Z. Huang. 2016. An efficient cloud market mechanism for computing jobs with soft deadlines. IEEE/ACM Trans. Network. 25, 2 (2016), 793–805.

[5] Testimonials and Case Studies. Retrieved November 20, 2022 from https://aws.amazon.com/ec2/spot/testimonials/.

[6] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafir. 2013. Deconstructing Amazon EC2 Spot Instance Pricing. ACM Trans. Econ. Comput. 1, 3 (2013), 20 pages.

[7] D. Ardagna, G. Casale, M. Ciavotta, J. F Pérez, and W. Wang. 2014. Quality-of-service in cloud computing: Modeling techniques and their applications. J. Internet Serv. Appl. 5(1) (2014), 1–17.

[8] J. Anselmi, D. Ardagna, John C. S. Lui, Adam Wierman, Y. Xu, and Z. Yang. 2017. The economics of the cloud. ACM Trans. Model. Perf. Eval. Comput. Syst. 2, 4, Article 18 (December 2017), 23 pages.

[9] V. Abhishek, I. Kash, and P. Key. 2012. Fixed and market pricing for cloud services. In Proceedings of the 7th Workshop on the Economics of Networks, Systems and Computation (NetEcon’12), IEEE, 157–162.

[10] L. Dierks, and S. Seuken. 2019. Cloud pricing: The spot market strikes back. In Proceedings of the ACM Conference on Economics and Computation (EC’19), ACM, 593–593.

[11] L. Dierks and S. Seuken. 2022. Cloud pricing: The spot market strikes back. Manage, Sci, 68, 1 (2022), 105–122.

[12] C. Kilcioglu and C. Maglaras. 2015. Revenue maximization for cloud computing services. SIGMETRICS Perform. Eval, Rev. 43, 3 (2015), 76.

[13] S. Chen, K. Moinzadeh, and Y. Tan. 2021. Discount schemes for the preemptive service of a cloud platform with unutilized capacity. Inf. Syst. Res. 32, 3 (2021), 967–986.

[14] X. Wu, F. De Pellegrini, G. Gao, and G. Casale. 2019. A framework for allocating server time to spot and on-demand services in cloud computing. ACM Trans. Model. Perf. Eval. Comput. Syst. 4, 4, Article 20 (2019), 31 pages.

[15] J. Song and R. Guérin. 2020. Pricing (and bidding) strategies for delay differentiated cloud services. ACM Trans. Econ. Comput. 8, 2, Article 8 (May 2020), 58 pages.

[16] S. Boopathians, F. Fusco, S. Leonardi, Y. Mansour, and R. Mehta. 2020. Online revenue maximization for server pricing. In Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence (IJCAI’20), 4106–4112.

[17] S. K. Garg, A. Toosi, S. K. Gopalaiyengar, and R. Buyya. 2014. SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J. Netw. Comput. Appl. 45 (2014), 108–120.

[18] K. Psychas and J. Ghaderi. On non-preemptive VM scheduling in the cloud. In Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS’17), 1, 2, Article 35 (2017), 29 pages.

[19] W. Dargie. 2014. Estimation of the cost of VM migration. In Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN’14), IEEE, 1–8.

[20] I. Kash and P. Key. 2016. Pricing the cloud. IEEE Internet Comput. 20, 1 (2016), 36–43.

[21] S. Yi, D. Kondo, and A. Andrzejak. Reducing costs of spot instances via checkpointing in the Amazon elastic compute cloud. In Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD’10), IEEE Computer Society, Los Alamitos, CA, 236–243.

[22] J. C. S. Kadupitiwe, V. Jad hao, and P. Sharma. Modeling the temporally constrained preemptions of transient cloud VMs. In Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing (HPDC’20), ACM, New York, NY, 41–52.

[23] N. Nisan, T. Roughgarden, Éva Tardos, and V. Vazirani. 2007. In Algorithmic Game Theory, Cambridge University Press.

[24] N. R. Devanur. 2017. A report on the workshop on the economics of cloud computing. ACM SIGecom Exch. 15.2 (2017), 25–29.

[25] X. Wu, P. Loiseau, and E. Hyytiä. 2020. Towards designing cost-optimal policies to utilize IaaS clouds with online learning. IEEE Transactions on. Parallel Distrib. Syst. 31, 3 (2020), 501–514.

[26] D. J. Dubois and G. Casale. 2016. OptiSpot: Minimizing application deployment cost using spot cloud resources. Cluster Comput. 19, 2 (2016), 893–909.

[27] X. Wu, H. Yu, G. Casale, and G. Gao. 2021. Towards cost-optimal policies for DAGs to utilize IaaS clouds with online learning.arXiv:2106.01847. Retrieved from https://arxiv.org/abs/2106.01847.

[28] Y. Azar, I. Kalp-Shaltiel, B. Lucier, I. Menache, J. Naor, and J. Yaniv. 2015. Truthful online scheduling with commitments. In Proceedings of the 16th ACM Conference on Economics and Computation (EC’15), ACM, 715–732.

[29] N. Jain, I. Menache, J. Naor, and J. Yaniv. 2015. Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters. ACM Trans. Parallel Comput. 2, 1 (2015), 3.

[30] X. Wu and P. Loiseau. 2015. Algorithms for scheduling deadline-sensitive malleable tasks. In Proceedings of the 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton’15), IEEE, 530–537.

[31] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau. 2015. Online auctions in IaaS clouds: Welfare and profit maximization with server costs. In Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS’15), ACM, 3–15.
