Report Number:

2017-05-01

Pricing and Bidding Strategies for Cloud Computing Spot Instances

Jiayi Song and Roch A. Guérin

We consider a cloud service based on spot instances and explore bidding and pricing strategies aimed at optimizing users' utility and provider's revenue, respectively. Our focus is on jobs that are heterogeneous in both valuation and sensitivity to execution delay. Of particular interest is the impact of correlation in these two dimensions. We characterize optimal bidding and pricing strategies under some simplifying assumptions, and more importantly highlight the impact of correlation in determining the benefits of a spot service over an on-demand service. We also provide a preliminary assessment of the results' robustness under more general assumptions.
... Read complete abstract on page 2.

Pricing and Bidding Strategies for Cloud Computing Spot Instances

Jiayi Song and Roch A. Guérin

Complete Abstract:

We consider a cloud service based on spot instances and explore bidding and pricing strategies aimed at optimizing users' utility and provider's revenue, respectively. Our focus is on jobs that are heterogeneous in both valuation and sensitivity to execution delay. Of particular interest is the impact of correlation in these two dimensions. We characterize optimal bidding and pricing strategies under some simplifying assumptions, and more importantly highlight the impact of correlation in determining the benefits of a spot service over an on-demand service. We also provide a preliminary assessment of the results' robustness under more general assumptions.

Pricing and Bidding Strategies for Cloud Computing Spot Instances

Jiayi Song and Roch Guérin
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, Missouri 63130
Email: (jiayisong,guerin)@wustl.edu

***Abstract*—We consider a cloud service based on spot instances and explore bidding and pricing strategies aimed at optimizing users' utility and provider's revenue, respectively. Our focus is on jobs that are heterogeneous in both valuation and sensitivity to execution delay. Of particular interest is the impact of correlation in these two dimensions. We characterize optimal bidding and pricing strategies under some simplifying assumptions, and more importantly highlight the impact of correlation in determining the benefits of a spot service over an on-demand service. We also provide a preliminary assessment of the results' robustness under more general assumptions.**

I. Introduction

Cloud computing has experienced explosive growth and become the computing platform of choice for an increasingly diverse set of users. Cloud providers have responded to this growing diversity by offering different types of computing services, each with its own pricing scheme[1].

In particular, Amazon offers three main pricing options that provide different trade-offs between the level of commitment the user is willing to make and the cost of the service. *On-demand instances* let users dynamically acquire compute capacity one hour at the time without any prior commitment, but come at a premium. In contrast, *reserved instances* call for a one or three year commitment, but enjoy a significant discount compared to on-demand instances. Finally, *spot instances,* like on-demand instances come with a one hour time granularity and do not require any type of commitment, but while they can be cheaper than on-demand instances, they exhibit large price variations (new spot instance prices are announced every hour). Users who have registered a *bid* in excess of the spot price gain access to the desired resource for the next hour[2], but lose it as soon as their bid falls below the next announced price. Amazon advertises historical spot price information for the past 90 days to enable users to devise bidding strategies that realize different trade-offs between cost and the ability to secure the desired resources. It also offers a tool, Spot Bid Advisor[3], which seeks to help users identify the right bidding level given a certain tolerance for service interruption.

[1]See https://aws.amazon.com/ec2/pricing for an example.

[2]Amazon recently introduced a variation of the spot service, Spot blocks, which lets a user specify a fixed duration for its spot instance. Once started a spot block runs uninterrupted for its stated period, but won't benefit of decreases in spot price during that time.

[3]https://aws.amazon.com/ec2/spot/bid-advisor/.

This paper is concerned with a cloud computing offering based on the spot service option. In other words, we consider users interested in the spot service of a cloud provider either as their main source of computing services, or as a backup/overflow facility for their own compute resources. Our goal is to develop a better understanding of user bidding strategies and provider pricing strategies. Of particular interest are pricing strategies that maximize the provider's revenue in the presence of users with diverse profiles. Users diversity can clearly take many different forms, and we focus on two core aspects with a direct influence on a user's willingness to pay, namely, job value and timeliness of job completion or sensitivity to delays caused by service interruptions. The value of a compute job is obviously of relevance when it comes to determining what a user is willing to pay to have it executed. Potentially more interesting in the context of a spot service is a job's sensitivity to any delay in its execution, as the latter is directly affected by the user's bidding strategy, *i.e.,* high bids ensure immediate execution while low bids are more likely to be interrupted multiple times and to incur large completion delays. In such an environment, an important question for a provider seeking to maximize its revenue, is how to account for differences in job valuation and sensitivity to delay across users when setting spot prices.

Pricing is typically a function of both offer (provider's capacity) and demand (user). In this paper, we assume, as others have recently done [6], that the provider capacity is large enough (infinite) to accommodate any demand, so that pricing is solely to maximize revenue given the user demand. This assumption is not unreasonable given the size of modern cloud computing facilities, and the fact that powered-down servers can be quickly brought online when needed [7], [8]. In addition, recent empirical work [3] analyzing Amazon's own pricing strategies hints at spare capacity that typically exceeds the demand. Hence, further validating the assumption that capacity constraints are rarely if ever present in modern public cloud systems.

Under these assumptions, our focus is on exploring how a cloud provider should set spot prices given a user population with heterogeneous profiles, *i.e.,* job valuations and sensitivity to delay. Individual user profiles are private information, but the cloud provider has knowledge of the profiles' distribution over the user population. Conversely, users are aware of past

spot prices, as in the Amazon EC2 setting, and can therefore rely on an empirical distribution of spot prices when making bidding decisions, *i.e.,* select bidding prices that optimize their own utility. In that context and consistent with the findings of [3], we assume that the provider selects a set of spot prices $\boldsymbol{p}$ and a corresponding distribution $\boldsymbol{\pi}$ for announcing prices so as to maximize its revenue.

Our investigation reveals several interesting features. Under reasonable assumptions regarding users' utility and the relation between a job's value and its execution time, we show that for any provider pricing, a fixed bidding strategy is optimal for users that decide to bid. In other words, a user either does not bid (the service is not cost-effective for her), or selects a bidding price, function of the job value and sensitivity to delay, and repeatedly bids at that price until the job completes. Conversely, we identify conditions under which a spot service yields no benefit to the provider over an on-demand service, *i.e.,* a service with a single spot price. Of particular interest is the fact that correlation between job value and sensitivity to delay plays a key role. Specifically, offering a spot service benefits a provider's revenue only when this correlation exceeds a certain threshold. The findings can help providers determine when a spot service is a meaningful addition to their offering, and users how to bid for spot instances.

The remainder of this paper is structured as follows. Section II briefly reviews previous works of relevance. Section III introduces our model more formally. Section IV investigates the user's bidding strategies, while Section V explores the provider's pricing strategies. The robustness of the results are tested numerically under more general assumptions in Section VI. Section VII summarizes the paper's findings.

II. Related Works

In this section, we review the vast literature on pricing in computing systems and highlight a few recent relevant works.

The idea of using pricing for resource allocation in computing systems with jobs that are heterogeneous in either value or sensitivity to delay is not new. It dates back to the 1960's with pricing for shared computing time *e.g.,* [5], [12]. In this early context, computing resource were typically constrained, with pricing used to realize an allocation of resources that maximized a global utility function across heterogeneous users. The finite resource assumption naturally lends itself to a queueing system formulation very different from that of this paper. The 1990 paper by Mendelson and Whang [9] offers a representative example. It considers an M/M/1 queue with multiple classes of jobs with different valuations and sensitivity to delay, and investigates pricing policies that maximize utility (social welfare) across classes. Conversely, [2] considers the provider's revenue maximization problem, and demonstrates that the damaged goods strategy of [4] also applies in this context.

However, while ideas of pricing regularly surfaced in the academic computing literature, their use in practice was limited [10] as pricing was never really necessary for the continued development and operation of large computing systems. Most of such systems were centrally controlled, *e.g.,* by the organization running the mainframe, which made defining usage policies easy, so that the complexity and cognitive overhead of mechanisms that price resources were not worth the payoff. Thus, research focused on scheduling algorithms to maximize utilization of shared resources rather than the pursuit of pricing policies to achieve explicit social goals [10].

The emergence of the cloud, with computing as a utility, changed this calculus. Users became accustomed to thinking about payoffs for timely job completions, with pricing an integral part of this assessment. In this context, two works close to this paper are [1], [6]. They target cloud computing services under a (mostly) infinite capacity setting while seeking to understand how job value and sensitivity to delay affect cloud service offerings. Both works offer interesting initial insight, but leave several questions unanswered, in particular regarding the role of correlation between a job's value and its sensitivity to delay. As we shall see, it plays an important role in determining to what extent a spot service can add value to an on-demand service offering[4].

III. Model Formulation

We assume a setting with a single cloud provider, *i.e.,* a monopoly environment, where the provider has access to "infinite" compute resources. The provider offers spot service with spot prices drawn from a set of n prices $\mathbf{p} = (p_1, p_2, ..., p_n), p_1 < p_2 < ... < p_n$, with a probability distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, ...\pi_n)$. Spot prices are updated periodically by randomly selecting a price from $\boldsymbol{p}$ according to $\boldsymbol{\pi}$. As in the Amazon EC2 spot service, users whose bid equals or exceeds the spot price are allowed to execute during that period, but their execution is stopped whenever their bid price falls below the spot price.

Demand for computing services is heterogeneous and originates from a large user population. Each job is characterized by its total computation time or length $t > 0$, its value v per unit of computation time, *i.e.,* the value of a job of length t is vt, and its sensitivity to computation delays $\kappa \geq 0$. Assume that customers know t, v and κ for all their jobs. For notation purposes, a job is represented through its triplet (t, v, κ). Job lengths are assumed independent of their unit value v and sensitivity to delay κ, and are drawn from a probability distribution with density function[5] $f(t)$. Note that the independence of v and t implies that a long job with a small v can be less valuable than a short job with a large v. Conversely, jobs' value and sensitivity to delay are drawn from a distribution with joint density function $q(v, \kappa)$. Hence, we allow for correlation between v and κ, *e.g.,* high value jobs can be more sensitive to delay (positive correlation).

Users are aware of $\boldsymbol{p}$ and $\boldsymbol{\pi}$, *e.g.,* from data published by the cloud provider (as with Amazon spot instances), and use

[4]We note that the result of [1] that an on-demand service is always superior, appears to be a consequence of its assumptions of finite spot service capacity and auction based mechanism to decide which jobs access that capacity.

[5]For simplicity, we assume that t, v, and κ are continuous random variables. Similar expressions are readily available for discrete random variables.

this information to select bidding strategies for new jobs. A bidding strategy Γ specifies both a first bidding price, as well as bids at subsequent bidding instances. These may in turn depend on past bids and spot prices, *i.e.,* Γ can be as simple as bidding at the same price until the job completes, or a complex state-dependent strategy that accounts for previous winning bids. Specifically, given spot prices $\boldsymbol{p}$ and associated distribution $\boldsymbol{\pi}$, a user selects among bidding strategies $\Gamma_{\boldsymbol{p},\boldsymbol{\pi}}(t, v, \kappa)$ for a job (t, v, κ) (denoted simply as Γ or $\Gamma(t, v, \kappa)$ when unambiguous), so as to maximize the job's *expected utility*, where expectation is computed over spot prices realizations. When the maximum expected utility is not positive, the user's bidding strategy is to not bid. In this case, the expected utility is taken to be zero (0). Otherwise, the job's expected utility under bidding strategy Γ is of the form

$$U(t, v, \kappa, \Gamma) = vt - P(t, v, \kappa, \Gamma) - D(t, v, \kappa, \Gamma),$$

where vt is the value derived from completing the job, $P(t, v, \kappa, \Gamma)$ is the expected execution cost of the job under bidding strategy Γ, and $D(t, v, \kappa, \Gamma)$ is the expected penalty associated with the job's execution delay under Γ. Additionally, a job's value is accrued only after it has completed, *i.e.,* there is no partial value for incomplete jobs.

Given $\boldsymbol{p}, \boldsymbol{\pi}$ and Γ, a user can readily compute the expected execution cost $P(t, v, \kappa, \Gamma)$ for a job of length t. Similarly, she can compute the expected job completion time $T(t, v, \kappa, \Gamma)$ so that $T(t, v, \kappa, \Gamma) - t$ represents the delay penalty above and beyond the job's execution time t under bidding strategy Γ, *e.g.,* the delay penalty is 0 when bidding at p_n, while it is maximum when bidding at p_1. Intuitively, $D(t, v, \kappa, \Gamma)$ should be an increasing function of κ and $T(t, v, \kappa, \Gamma) - t$. There are many possible choices for such a function, and in Section VI we experiment with concave and convex increasing functions. However, for analytical tractability, we assume a linear function of the form $\kappa\,(T(t, v, \kappa, \Gamma) - t)$. This then yields a utility function of the form

$$U(t, v, \kappa, \Gamma) = vt - P(t, v, \kappa, \Gamma) - \kappa(T(t, v, \kappa, \Gamma) - t). \quad (1)$$

Selecting an optimal bidding strategy Γ^* for a job (t, v, κ), therefore, consists of solving the following problem:

$$\Gamma^*(t, v, \kappa) = \arg\max_{\Gamma} U(t, v, \kappa, \Gamma), \quad (2)$$

where we further make the assumption that if a customer starts bidding for a job, she continues bidding for it until it completes. In other words, once a user identifies an optimal strategy that yields a positive expected utility, she does not revisit her decision and proceeds with this strategy until the job completes. Note that while this ensures a positive average (over a large number of jobs) utility, it is possible for individual jobs to realize a negative utility, *e.g.,* when encountering a long sequence of high spot prices. This is the price paid for the simplicity of not re-evaluating bidding strategies in every period. We explore numerically in Section VI the impact of allowing bidding to terminate once the expected utility associated with further bids becomes negative.

The previous discussion identified how, given a known distribution of spot prices as characterized by $\boldsymbol{p}, \boldsymbol{\pi}$, a user selects an optimal bidding strategy for a job (t, v, κ). Conversely, given such a behavior on the part of users, a natural question is how the cloud provider should select $\boldsymbol{p}$ and $\boldsymbol{\pi}$ so as to maximize its expected per job revenue $R_{\boldsymbol{p},\boldsymbol{\pi}}$, *i.e.,* solve

$$(\boldsymbol{p}^*, \boldsymbol{\pi}^*) = \arg\max_{\boldsymbol{p},\boldsymbol{\pi}} R_{\boldsymbol{p},\boldsymbol{\pi}}, \quad (3)$$

where $R_{\boldsymbol{p},\boldsymbol{\pi}}$ is given by

$$R_{\boldsymbol{p},\boldsymbol{\pi}} = \iiint_{t,v,\kappa} f(t)q(v, \kappa)P\left(t, \Gamma^*_{\boldsymbol{p},\boldsymbol{\pi}}(t, v, \kappa)\right) dt\, dv\, d\kappa.$$

In solving the optimization of Eq. (3), we assume that the triplets (t, v, κ) are private information, but that the cloud provider has knowledge of $f(t)$ and $q(v, t)$, *e.g.,* from empirical customer data.

In the next two sections, we proceed to characterize first Γ^* and then $(\boldsymbol{p}^*, \boldsymbol{\pi}^*)$. Due to space constraints, we omit all proofs, which can be found in [11].

IV. Optimal Bidding Strategy

In this section, we show that for all jobs and all pricing systems an optimal fixed bidding strategy exists that only depends on κ. We assume that customers only consider pure strategies, *i.e.,* never choose a mixed bidding strategy. This assumption is reasonable as an optimal mixed bidding strategy is simply a randomization among several pure optimal bidding strategies that have the same expected utility as the optimal pure strategy.

Since customers who choose to bid do not terminate jobs before completion, Eq. (1) implies that maximizing utility is equivalent to minimizing expected cost (vt is constant), which for job (t, v, κ) under strategy Γ is of the form:

$$C(t, v, \kappa, \Gamma) = P(t, v, \kappa, \Gamma) + \kappa(T(t, v, \kappa, \Gamma) - t), \quad (4)$$

where recall that $P(t, v, \kappa, \Gamma)$ is the expected payment, and $T(t, v, \kappa, \Gamma)$ is the job's expected completion time. Both $P(t, v, \kappa, \Gamma)$ and $T(t, v, \kappa, \Gamma)$ are expectations over realizations of spot prices and bidding prices under Γ.

To simplify notation, when focusing on a specific job, we use $C(\Gamma)$ to denote the expected cost under strategy Γ. For ease of presentation, we also assume that job lengths are integer multiple of the slot length, and w.l.o.g. assume the slot length to be 1. We narrow the strategy space to those with bidding prices in $\{p_1, ..., p_n\}$. This does not affect the optimal strategy, as shown in the following lemma.

Lemma 1. *Bidding at a price $b \in [p_i, p_{i+1})$, where $1 \le i \le n$, generates the same expected cost and the same winning probability as bidding at p_i.*

We first prove the existence of an optimal fixed bidding strategy for jobs of length one-slot.

Proposition 2. *For job $(1, v, \kappa)$ and any pricing system, there always exists an optimal bidding strategy.*

The proof of Proposition 2 (see [11]) argues that if there is no optimal strategy, an infinite sequence of strategies exists whose expected cost approaches (but never reaches) the infimum, and proceeds to show that the infimum can be achieved by an optimal fixed bidding strategy. This establishes the existence of an optimal strategy and hints at the fact that it is a fixed bidding strategy. However, given the existence of an optimal bidding strategy, the strategy space need not be continuous, so that the next proposition is necessary to formally establish the optimality of a fixed bidding strategy.

Proposition 3. *For job $(1, v, \kappa)$, there exists an optimal fixed bidding strategy.*

The next proposition extends the optimality of a fixed bidding strategy to jobs of arbitrary length.

Proposition 4. *A strategy that bids at $b^*(v, \kappa)$ in every slot is an optimal bidding strategy for (t, v, κ), where $b^*(v, \kappa)$ is the optimal bidding price for $(1, v, \kappa)$.*

Proposition 4 establishes not only that a fixed bidding strategy is optimal, it also shows that the optimal bidding price $b^*(v, \kappa)$ is independent of the job length. The next step is to characterize $b^*(v, \kappa)$, or more precisely, as we shall see, $b^*(\kappa)$, *i.e.,* the optimal bidding price only depends on a job's sensitivity to delay. However, note that whether or not a customer bids for a job depends on the job value v. In characterizing $b^*(\kappa)$, we also show that computing the optimal bidding price can be realized with a simple linear search.

If a customer bids for a job (t, v, κ) using a fixed bidding strategy with a bidding price b, the expected fraction of time or probability that the job is active is

$$0 \leq \alpha(b) = \sum_{p_i \leq b} \pi_i \leq 1. \tag{5}$$

The expected payment per unit of time *given* that the job is active is then

$$p(b) = \frac{\sum_{p_i \leq b} \pi_i p_i}{\alpha(b)}, \tag{6}$$

so that the expected payment is

$$P(t, b) = p(b)t. \tag{7}$$

and the average job completion time is

$$T(t, b) = \frac{t}{\alpha(b)}. \tag{8}$$

Denote the expected cost for the strategy that bids at b as $C(t, v, \kappa, b)$ Substituting Eq. (7) and Eq. (8) in Eq. (4) gives

$$C(t, v, \kappa, b) = -t\left(p(b) + \kappa\left(\frac{1}{\alpha(b)} - 1\right)\right). \tag{9}$$

Note that for a given b, $C(t, v, \kappa, b)$ is proportional to t (as is $U(t, v, \kappa, b)$). Note also that the optimal bidding price needs not to be unique. W.l.o.g, assume that users always bid at the lowest optimal price. The optimal fixed bidding price is then

$$\begin{aligned} b^*(v, \kappa) &= \min \arg\min_{i \in \{1,2,\ldots n\}} C(t, v, \kappa, p_i) \\ &= \min \arg\min_{i \in \{1,2,\ldots n\}} C(1, v, \kappa, p_i), \end{aligned}$$

where the second equality comes directly from $C(t, v, \kappa, b)$'s proportionality to t (and/or Proposition 4).

Proposition 5. *A customer's optimal fixed bidding price $b^*(\kappa)$ is independent of v and t, and non-decreasing in κ. Specifically, a customer with $\kappa \in (\kappa_{i-1}, \kappa_i]$ will bid at p_i if she chooses to adopt the service, where $\kappa_i = \sum_{j \leq i}(p_{i+1} - p_j)\pi_j$ for $i \geq 1$, and $\kappa_0 = -\epsilon < 0$.*

Proposition 5 states two interesting results: 1) valuation plays no role in the optimal bidding price; and 2) the optimal bidding price is an increasing function of κ. Note though that while v does not influence the choice of a bidding price, it affects whether to bid or not. Additionally, as we shall see in Section V, the distribution of v also plays a role in the choice of the service provider's pricing strategy. We also note that 2) is intuive as a higher delay sensitivity implies a greater willingness to pay to avoid delays.

Finally, we show how a job's optimal bidding price can be obtained using a simple linear search.

Corollary 6. *A job's optimal bidding price can be determined using a simple linear search.*

The corollary's proof is constructive in nature and takes advantage of the structure of a job's utility characterized in Proposition 5, which shows that it admits a single maximum as a function of the bidding price.

V. Optimal Pricing Strategy

In this section, we turn to characterizing how a cloud service provider should price its spot service to maximize revenue given that users bid according to the optimal bidding strategy of the previous section.

For simplicity, we limit ourselves to binary job profiles with only two job valuations $(0 < v_1 < v_2)$ and delay sensitivity $(0 \leq \kappa_1 < \kappa_2)$. While obviously a simplification, this still captures job heterogeneity along two dimensions, and allows us to incorporate correlation between those dimensions. As we shall see, the latter plays an important role in the structure of the optimal pricing configuration.

We start with a simple result that limits the number of prices the cloud provider needs to consider under those assumptions.

Lemma 7. *For binary profiles with $0 \leq \kappa_1 < \kappa_2$, the optimal pricing system needs at most two prices.*

Denote the prices as p_1 and p_2, where $0 < p_1 < p_2$, and denote the probability of p_1 being selected as π. Our goal is to characterize p_1, p_2, and π, as functions of $v_1, v_2, \kappa_1, \kappa_2$ and the correlation between them. We start with a configuration where v and κ are independent.

A. Independent v and κ

We assume that job valuations and delay sensitivities are independent, with a job having valuation v_1 with probability p and delay sensitivity κ_1 with probability q, *i.e.,*

	κ_1	κ_2	
v_1	pq	$p(1-q)$	p
v_2	$q(1-p)$	$(1-p)(1-q)$	$1-p$
	q	$1-q$	

We show that under these assumptions, a single spot price maximizes revenue, *i.e.,* the optimal spot pricing strategy is equivalent to an on-demand service.

Proposition 8. *For a system with independent v and κ, where $v \in \{v_1, v_2\}, 0 < v_1 < v_2$ and $\kappa \in \{\kappa_1, \kappa_2\}, 0 \leq \kappa_1 < \kappa_2$, one-price service maximizes the expected revenue. Specifically,*

- *if $v_1 > v_2(1-p)$, a price of $v_1 - \epsilon$ maximizes revenue and realizes full adoption;*
- *if $v_1 \leq v_2(1-p)$, a price of $v_2 - \epsilon$ maximizes revenue, and only customers with valuation v_2 will adopt.*

Proposition 8 seems counterintuitive as market segmentation usually leads to higher revenue. However, under a two-price spot service, we basically ask jobs with large delay sensitivity to pay more. This in turn has the potential to 1) exclude jobs with large delay sensitivity and small valuation, and 2) extract a smaller price from jobs with small delay sensitivity and large valuation. Keeping both quantities small is key to generating a higher revenue, and this ends-up not being feasible when job valuation and delay sensitivity are independent. In the next section, we explore how this result changes as correlation between the two varies.

B. Correlated v and κ

We begin our investigation with the cases of perfect negative or positive correlation between job valuation and delay sensitivity, and then proceed with general correlation.

Proposition 9. *When valuation and delay sensitivity are perfectly negatively correlated,* i.e., *the system only has jobs (v_1, κ_2) and (v_2, κ_1), where $0 < v_1 < v_2$ and $0 \leq \kappa_1 < \kappa_2$, a single price spot service is optimal,* i.e., *maximizes revenue.*

The result is reasonably intuitive as a two-price system with a low enough price p_1 to attract (v_1, κ_2) jobs will also have (v_2, κ_1) find p_1 attractive. Next, we consider the case of perfect positive correlation.

Proposition 10. *When valuation and delay sensitivity are perfectly positively correlated,* i.e., *the system only has jobs (v_1, κ_1) and (v_2, κ_2), where $0 < v_1 < v_2$ and $0 \leq \kappa_1 < \kappa_2$, then using q to denote the fraction of (v_1, κ_1) jobs, we have*

- *When $\kappa_2(1-q) - \kappa_1 > 0$ and $v_1\kappa_2 > v_2\kappa_1$, a two-price spot service is optimal;*
- *Otherwise, a one-price spot service is optimal.*

Note that $v_1\kappa_2 > v_2\kappa_1$ is equivalent to $\frac{\kappa_2}{\kappa_1} > \frac{v_2}{v_1}$ when $v_1, \kappa_1 > 0$. Hence, the proposition states that when the relative difference in delay sensitivities is larger than that of valuations *and* the fraction of (v_2, κ_2) jobs is large, a two-price spot service can generate a higher revenue.

Next, we turn to the general case of intermediate correlation. For that purpose, we consider distributions with fixed marginals. Specifically, v_1 has a fixed marginal a and κ_1 has a fixed marginal b, where $a, b \in (0, 1)$. Denote the probability that a job is of type (v_i, κ_j) as q_{ij}, where $i, j \in \{1, 2\}$. Then

	κ_1	κ_2	
v_1	q_{11}	q_{12}	a
v_2	q_{21}	q_{22}	$1-a$
	b	$1-b$	

The next proposition characterizes the optimal pricing strategy in this configuration. In particular, it highlights the presence of a possible correlation threshold ρ^*.

Proposition 11. *Given a distribution with fixed marginals for job valuation and delay sensitivity, depending on the value of ρ^*, where ρ^* is a function of $a, b, \kappa_1, \kappa_2, v_1$ and v_2, the optimal pricing strategy takes either one of two forms as the correlation coefficient ρ varies from -1 to 1:*

- *if $\rho^* \in (-1, 1)$, when $\rho < \rho^*$, a one-price spot service is optimal and when $\rho \geq \rho^*$, a two-price spot service is optimal;*
- *otherwise, a one-price spot service is always optimal whatever ρ.*

An explicit expression for ρ^* is available in the proof of Proposition 11. Note also that given Proposition 8, Proposition 11 implies that a one-price spot service, *i.e.,* an on-demand service, is optimal in all configurations with a non-positive correlation between valuation and delay sensitivity.

VI. Robustness Evaluation

This section offers a preliminary assessment of the extent to which the findings of the previous section remain valid under more general conditions. Because the system becomes quickly intractable as simplifying assumptions are relaxed, the investigation is carried out using numerical analysis. Our focus is on testing the role of correlation, in particular the presence of ρ^*, when assumptions are relaxed.

A. Allowing Job Terminations

In this section, we allow customers to terminate jobs under certain conditions. The bidding strategy remains the optimal fixed bidding strategy of Section IV, but customers terminate jobs once their residual expected utility is no longer positive. Specifically, for a job (t, v, κ) that started at time 0 with an optimal bidding price p_i, let $t_0 < t$ denote the execution time of the job up to time T_0. The job terminates at T_0 if

$$vt - p(p_i)(t - t_0) - \kappa \left(T_0 + \frac{t - t_0}{\alpha(p_i)} - t \right) \leq 0 \,, \qquad (10)$$

vt is the value that successfully completing the job would yield, $p(p_i)(t-t_0)$ is the expected additional cost for completing the job, and $\kappa \left(T_0 + \frac{t-t_0}{\alpha(p_i)} - t \right)$ is the job's total expected delay penalty. Eq. (10) states that the job will be terminated, once its expected utility going forward becomes negative.

Eq. (10) also points out that termination decisions depend on job length. To test this factor, we choose a job length distribution, where jobs have length 1 with probability r (we

consider different values for r) and length 5 otherwise, while reusing the binary job profile of Section V. We fix v_1 and κ_1 to 0.1, and independently vary v_2 and κ_2 from 0.1 to 1. We also vary the distribution marginals a and b. Because of the complexity of characterizing the optimal pricing strategy, we assume that the service provider is oblivious to the fact that customers may terminate jobs, *i.e.,* it sets prices ignoring job terminations.

Numerical results highlight that Proposition 11 holds, but only when at least one of a or b is close to either 0 or 1, *i.e.,* the distribution exhibits a strong mode, while r does not have much influence. This likely stems from the assumption that pricing remains oblivious to possible job terminations. Specifically, revenue differences between termination and non-termination can be large when job profiles are evenly distributed, as the service provider needs to balance between all job types. It is, therefore, of interest to investigate the structure of the optimal pricing strategy when job termination is allowed, and to evaluate whether Proposition 11 now holds more broadly. This is a topic for future work.

B. Convex and Concave Delay Sensitivity Functions

Our model assumes linear sensitivity to delay. We relax this assumption using convex and concave piecewise linear functions. As before, we use a binary job profile with jobs of length 1 with probability r and of length 5 otherwise. The pricing policy is, however, now (numerically) obtained for the new delay sensitivity functions.

1) Convex Delay Sensitivity: A job's delay sensitivity is of the form:

$$D_1(\kappa, t) = \max\{0, \kappa(T(t) - T^*)\},$$

$T(t)$ is the expected execution time, and T^* is a threshold. In other words, the job is insensitive to delay until T^* and experiences a linear penalty with slope κ after that. Note that the optimal bidding strategy now depends on κ, t and T^*.

We ran experiments with $r = 0.2, 0.5, 0.8$, and $T^* = 2, 6, 10$. We fixed v_1 to 0.1, and κ_1 to 0 and varied v_2 from 0.2 to 1, and κ_2 from 0.1 to 1. Proposition 11 held across all experiments. Of note is the fact that a two-price spot service was found to generate a higher revenue even in some configurations with negative correlation. This is somewhat intuitive as under $D_1(\kappa, t)$, small jobs with high delay sensitivity are essentially similar to small jobs with 0 delay sensitivity.

2) Concave Delay Sensitivity: A job's delay sensitivity is of the form:

$$D_2(\kappa, t) = \min\{\kappa(T(t) - t), \max\{0, \kappa(T^* - t)\}\},$$

In other words, the delay penalty initially increases linearly with slope κ, and then stays constant after T^*.

We ran experiments for the same configurations as $D_1(\kappa, t)$, and in most cases Proposition 11 still held. There were, however, a few counterexamples. In particular, $v_1 = 0.1, v_2 = 0.4, \kappa_1 = 0.1, \kappa_2 = 0.4, r = 0.8$, and $T^* = 2$ with both marginals set to 0.8, resulted in an optimal two-price spot service for $\rho = -0.25$, while a one-price spot service was optimal for $\rho = 0.375$. This may due to the fact that the initial expected delay penalty is scaled by the relatively large job length (80% of jobs have length 5), which complicates the optimal bidding strategies. Investigating this behavior further is of interest and the topic of future work. We also found that as T^* increases, the number of counterexamples decreases. This is expected since for large T^*, $D_2(\kappa, t)$ is increasingly similar to a linear function.

C. Continuous Distributions

Our investigation has been limited to four job types with binary distributions for v and κ. We test the results for a larger number of job types using uniform distributions. Marginals are kept fixed and correlation between job value and delay sensitivity is varied using a Gaussian copula.

Characterizing the optimal pricing strategy for continuous distributions is complex. For simplicity, we limit pricing to two prices, which is sufficient to test Proposition 11. We fix the minimum job valuation and delay sensitivity to $v_{\min} = 0.1$, and $\kappa_{\min} = 0$, respectively, and vary the maximum valuation and delay sensitivity, v_{max}, κ_{max}, from 0.1 to 1.5.

Fig. 1 reports results for $v_{\max} = 0.9$ and different values of $k_{\max}$. Similar results were obtained for other values of $v_{\max}$. The figure shows that Proposition 11 still holds, with ρ^* a decreasing function of $k_{\max}$. This is intuitive as a higher $k_{\max}$ corresponds to a larger number of delay sensitive jobs.

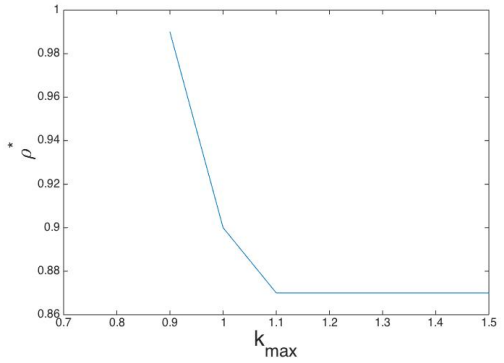

Fig. 1. ρ^* for continuous distributions and $v_{max} = 0.9$

VII. Conclusion

The paper explored the potential benefits (to users and cloud providers) of a spot service. It characterized optimal bidding and pricing strategies when jobs are heterogeneous in both valuation and sensitivity to execution delay. More importantly, it highlighted the role of correlation between job valuation and sensitivity to delay in determining whether a spot service is of value over an on-demand only service. In particular, it showed that a minimum level of correlation between job valuation and sensitivity to delay is necessary for a spot service to be valuable. An initial assessment of the results' robustness showed that they remain valid under more general conditions, though a complete investigation of the solution space remains a topic for future work.

REFERENCES

[1] V. Abhishek, I. A. Kash, and P. Key. Fixed and market pricing for cloud services. In *Proc. NetEcon'12*, Orlando, FL, March 2012.

[2] P. Afèche. Incentive-compatible revenue management in queueing systems: Optimal strategic delay. *Management & Service Operations Management*, 15(3):423–443, Summer 2013.

[3] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir. Deconstructing amazon EC2 spot instance pricing. *ACM Trans. Econ. Comput.*, 1(3), September 2013.

[4] R. J. Deneckere and R. P. McAfee. Damaged goods. *Journal of Economics & Management Strategy*, 5(2):149–174, Summer 1996.

[5] D. S. Diamond and L. L. Selwyn. Considerations for computer utility pricing policies. In *Proceedings of the 1968 23rd ACM National Conference*, pages 189–200. ACM, 1968.

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

[7] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, 12(1), 2009.

[8] M. Mao and M. Humphrey. A performance study on the VM startup time in the cloud. In *Proc. IEEE CLOUD 2012*, Honolulu, HI, June 2012.

[9] H. Mendelson and S. Whang. Optimal incentive-compatible priority pricing for the M/M/1 queue. *Operations Research*, 38(5):870–883, September-October 1990.

[10] J. Shneidman, C. Ng, D. C. Parkes, A. AuYoung, A. C. Snoeren, A. Vahdat, and B. N. Chun. Why markets could (but don't currently) solve resource allocation problems in systems. In *Proceedings of the 10th Workshop on Hot Topics in Operating Systems (HotOS)*, 2005.

[11] J. Song and R. Guerin. Pricing and bidding strategies for cloud computing spot instances. Technical report, Washington University in St. Louis, January 2017. Available at https://dl.dropboxusercontent.com/u/69358620/cloud_pricing%2Bproofs.pdf.

[12] I. E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, 1968.