Simple Pricing Schemes for the Cloud

Ian A. Kash[1], Peter Key[1], and Warut Suksompong[2(✉)]

[1] Microsoft Research, Cambridge, UK
{iankash,Peter.Key}@microsoft.com
[2] Department of Computer Science, Stanford University, Stanford, USA
warut@cs.stanford.edu

Abstract. The problem of pricing the cloud has attracted much recent attention due to the widespread use of cloud computing and cloud services. From a theoretical perspective, several mechanisms that provide strong efficiency or fairness guarantees and desirable incentive properties have been designed. However, these mechanisms often rely on a rigid model, with several parameters needing to be precisely known in order for the guarantees to hold. In this paper, we consider a stochastic model and show that it is possible to obtain good welfare and revenue guarantees with simple mechanisms that do not make use of the information on some of these parameters. In particular, we prove that a mechanism that sets the same price per time step for jobs of any length achieves at least 50% of the welfare and revenue obtained by a mechanism that can set different prices for jobs of different lengths, and the ratio can be improved if we have more specific knowledge of some parameters. Similarly, a mechanism that sets the same price for all servers even though the servers may receive different kinds of jobs can provide a reasonable welfare and revenue approximation compared to a mechanism that is allowed to set different prices for different servers.

1 Introduction

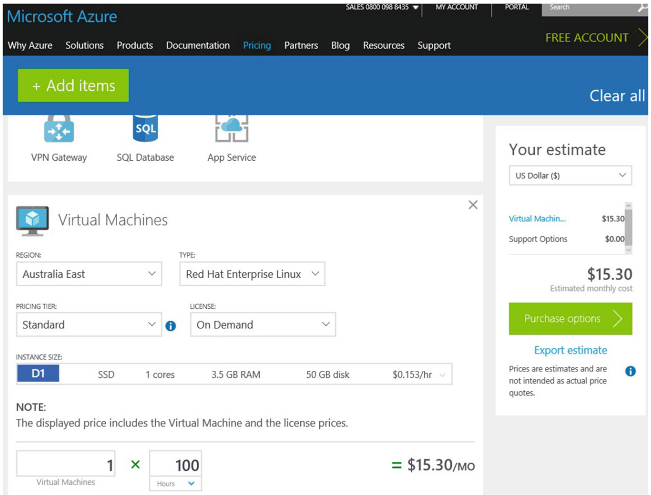
With cloud computing generating billions of dollars per year and forming a significant portion of the revenue of large software companies [10], the problem of how to price cloud resources and services is of great importance. On the one hand, for a pricing scheme to be used, it is necessary that the scheme provide strong welfare and revenue guarantees. On the other hand, it is also often desirable that the scheme be simple. We combine the two objectives in this paper and show that simple pricing schemes perform almost as well as more complicated ones with respect to welfare and revenue guarantees. In particular, consider the pricing scheme for virtual machines on Microsoft Azure shown in Fig. 1. Once the user chooses the basic parameters such as region, type, and instance size, the price is calculated by simply multiplying an hourly base price by the number of virtual machines and number of hours desired. The question that we study can be phrased in this setting as follows: How much more welfare or revenue

The full version of this paper is available at http://arxiv.org/abs/1705.08563.

© Springer International Publishing AG 2017
N. R. Devanur and P. Lu (Eds.): WINE 2017, LNCS 10674, pp. 311–324, 2017.
https://doi.org/10.1007/978-3-319-71924-5_22

could be created if instead of this simple multiplication formula, a complex table specifying the price for each number of hours were to be used? Our main result is that the former offers at worst a two approximation to the latter, both in terms of welfare and revenue. Similarly, we demonstrate that setting a single price for a group of servers, even though the servers may receive different kinds of jobs, can provide a reasonable welfare and revenue approximation compared to setting different prices for different servers.

In much of the prior work in this space, which focuses more explicitly on scheduling, prices depend in a complex way on a number of parameters (typically including job length, arrival time, deadline, and value) as well as the current state of the system [3,11,20,21,24]. A weakness of such schemes is that they require these parameters to be known up front in order for the desirable properties of the mechanisms, such as their approximation ratios, to hold. The availability of such information is not always realistic in practice. Even when it is in principle possible to provide this information, there is a cost to participants in both time and resources to figure it out. In this work, we show that good results are possible with no up front information.



Virtual Machines

REGION: Australia East

TYPE: Red Hat Enterprise Linux

PRICING TIER: Standard

LICENSE: On Demand

INSTANCE SIZE: D1 SSD 1 cores 3.5 GB RAM 50 GB disk $0.153/hr

NOTE:
The displayed price includes the Virtual Machine and the license prices.

1 Virtual Machines × 100 Hours = $15.30/MO

Your estimate

US Dollar ($)

Virtual Machin...	$15.30
Support Options	$0.00

$15.30
Estimated monthly cost

Prices are estimates and are not intended as actual price quotes.

Fig. 1. Pricing scheme for virtual machines on Microsoft Azure [4].

For our initial results we assume that there is a single server, which receives jobs of various lengths whose value per time step is drawn from the same probability distribution regardless of length. We compare the welfare and revenue that can be obtained by setting a price per time step that is independent of the job length against the corresponding objective obtained by setting an individual price for each job length. When we are allowed the freedom of setting different prices for different job lengths, intuitively we want to set a higher price per time step for longer jobs as a premium for reserving the server for a longer period of time.[1] However, as we show, we do not lose more than 50% of the welfare or revenue if we are only allowed to set one price. We would like to emphasize that this is a worst-case bound over a wide range of parameters, including the number of job lengths, the distribution over job lengths, and the distribution over job values. Indeed, as we show, we can obtain improved bounds if we know the value of some of these parameters. The price that we use in the single-price setting can be chosen from one of the prices used in the multi-price setting, meaning that we do not have to calculate a price from scratch. Moreover, all of our approximation guarantees hold generally for arbitrary prices, meaning that for any prices that we may set in a multi-price setting (i.e., not necessarily optimal ones), we can obtain an approximation of the welfare or revenue by setting one of those prices alone. Finally, we emphasize that these results put no restrictions on the form of the distribution; it can be discrete, continuous, or mixed. The only substantive constraint is that jobs of all lengths share the same distribution of value per time step. However, in an extension we show that a version of our results continues to hold even if this constraint is relaxed.

We then generalize our results to a setting where there are multiple servers, each of which receives jobs of various lengths. The distribution over job lengths can be different for different servers. This is conceivable, for instance, if the servers are in various geographic locations or are utilized by various groups of users. We compare the welfare and revenue obtained by a simple pricing scheme that sets the same price for all servers against the corresponding objective achieved by a scheme that can set a different (single) price for each server. Roughly speaking, we show that as long as the parameters are not too extreme, e.g., the number of servers or the job lengths are not too large, then we do not lose too much of the welfare or revenue by setting a single price. Combining this with our initial results, we obtain an approximation of a very restricted pricing scheme where we must set the same price for all servers and all job lengths against one where we can set an individual price for each job length of each server. These results require an assumption that all servers have the same probability of not receiving a job at a time step. Using similar techniques, we also obtain approximation bounds when this assumption does not hold but there is only one job length across all servers.

1.1 Related Work

Much recent work has focused on designing online scheduling mechanisms with good welfare guarantees and incentive properties. Jain et al. [20] exhibited a truthful mechanism for batch jobs on cloud systems where jobs are allocated non-preemptively, and the same group of authors came up with mechanisms for deadline-sensitive jobs in large computing clusters [21]. Lucier et al. [24] also considered the problem of scheduling deadline-sensitive jobs; they circumvented

[1] Amazon recently started offering a product called “defined duration spot instances” where users can specify a duration in hourly increments up to six hours [2]. Indeed, the price *per hour* of this product increases as the number of hours increases.

known lower bounds by assuming that jobs could be delayed and still finish by their deadline. Zhang et al. [26] developed a framework for truthful online cloud auctions where users with heterogeneous demands can come and leave on the fly. More recently, Azar et al. [3] constructed a truthful mechanism that achieves a constant competitive ratio given that slackness is allowed, while Dehghani et al. [11] assumed a stochastic model and developed a truthful mechanism that approximates the expected maximum welfare up to a constant factor. Wang et al. [25] designed mechanisms for selling reserved instances where users are allowed to reserve resources of any length and from any time point in the future. Other work in this space has dealt with comparing pricing mechanisms such as the on-demand market and the spot market [1,12,19], achieving fairness in job allocation [17], and studying models of real-time pricing with budget constraints [18]. Kash and Key [22] gave a survey of the current state of research in economics and computer science with respect to cloud pricing.

From a technical perspective, our work bears a resemblance to the work of Dütting et al. on discriminatory and anonymous posted pricing and of Disser et al. on hiring secretaries. In particular, Dütting et al. [14] considered the problem of selling a single item to buyers who arrive sequentially with values drawn independently from identical distributions. They showed that by posting discriminatory prices, one can obtain at most $2 - 1/n$ times as much revenue as that obtained by posting the same anonymous price, where n is the number of buyers. As is also the case in our work, their anonymous price can always be chosen from one of the discriminatory prices, but their bound is obtained via a relaxation of the discriminatory pricing problem, a different technique than what we use. Disser et al. [13] provided a competitive online algorithm for a variant of the stochastic secretary problem, where applicants need to be hired over time. When each applicant arrives, the cost per time step of the applicant is revealed, and we have to decide on the duration of the employment. Once an applicant is accepted, we cannot terminate the contract until the duration of the job is over.

Our work falls into the broader area of the design and analysis of simple mechanisms, particularly posted price mechanisms. One of the motivations for studying simple mechanisms is that in practice, designers are often willing to partially give up optimality in return for simplicity. Mechanisms that simply post prices on goods have received significant attention since they reflect perhaps the most common way of selling goods in the real world, and moreover they leave no room for strategizing, making them easy for agents to participate in. A long line of work has investigated how well such mechanisms can approximate optimal mechanisms with respect to various objectives including welfare [9,15,16], revenue [5–7], and social costs [8]. In Sect. 3.4 we show that techniques from this literature can recover some of our results under relaxed assumptions.

2 Preliminaries

We consider a system with a number of servers and discrete time steps. Each job takes an integer number of time steps to complete and yields a value upon completion. The value *per time step* of a job is drawn from a known distribution which is independent of the length of the job. Let F be the cumulative distribution function of this distribution and f the probability density function with respect to a base measure μ, and define $\ell(x) = xf(x)$.[2] We do not make any assumption on our distribution; in particular, it need not be continuous or discrete, which is why we allow flexibility in terms of the base measure.

When a job request is made for a job to be served by a server, there is a price p per time step which may depend on the job length and/or the server. If the value per time step of the job is at least p, the server accepts and executes the job to completion. Otherwise, the server rejects the job. The objectives in our model are the steady-state welfare and revenue for each pricing scheme. In particular, we will be interested in the expected welfare and revenue per time step, given that the job values are drawn from a probability distribution. This can also be thought of as the average welfare and revenue per time step that result from a pricing scheme over a long period of time.

In Sect. 3, we assume that there is a single server. Each time step, either zero or one job appears. A job with length a_i appears with probability $0 < r_i \leq 1$, where $\sum_{i=1}^{n} r_i \leq 1$ and n denotes the number of job lengths. We are allowed to set a price p_i for jobs of length a_i. If a server accepts a job of length a_i, it is busy and cannot accept other jobs for a_i time steps, including the current one. We compare the setting where we are forced to set the same price p for all job lengths against the setting where we can set a different price p_i for each job length a_i. Note that if we could set different prices for different job lengths, then to optimize welfare or revenue, intuitively we would set a higher price per time step for longer jobs as a premium for reserving the server for a longer period. Put differently, once we accept a longer job, we are stuck with it for a longer period, during which we miss the opportunity to accept other jobs. Consequently, we should set a higher standard for accepting longer jobs. (See also Footnote 1.)

In Sect. 4, we assume that there are multiple servers. Each time step, either zero or one job appears for each server $1 \leq j \leq n$. For server j, a job with length a_{ji} appears with probability $0 < r_{ji} \leq 1$ for $1 \leq i \leq n_j$, where n_j denotes the number of job lengths for server j. We do not assume that the set of job lengths or the number of job lengths are identical across servers. On the other hand, we assume that the probability of no job appearing at a time step is the same for all servers, i.e., $\sum_{i=1}^{n_j} r_{ji}$ is constant for any j. In Subsect. 4.1, we assume that we can set one price per server, and we compare the setting where we are forced to set the same price p for all servers against that where we can set a different price p_j for each server j. In Subsect. 4.2, we assume that we can set a different price p_{ji} for each server j and each of its job lengths a_{ji}, and we compare that

[2] For technical reasons, we will deviate slightly from the usual notion of cumulative distribution function. In particular, if y is a random variable drawn from a distribution, then we define its cumulative distribution function $F(x)$ as $\Pr[y < x]$ instead of the usual $\Pr[y \leq x]$. This will only be important when we deal with discrete distributions.

setting against that where we are forced to set the same price p for all servers and all job lengths.

All proofs can be found in the full version of this paper [23].

3 One Server

In this section, we assume that there is a single server, which receives jobs of various lengths. After presenting an introductory example in Subsect. 3.1, we consider the general setting with an arbitrary number of job lengths in Subsect. 3.2. In this setting, we show a 50% approximation for both welfare and revenue of setting one price for all job lengths compared to setting an individual price for each job length, for any realization of the parameters. Moreover, we show in Subsect. 3.3 that our techniques provide a template for deriving tighter bounds if we have more specific information on the parameters. In particular, when there are two job lengths, we show for each setting of the parameters a tight approximation bound for welfare and revenue. Our approximation results hold for arbitrary (i.e., not necessarily optimal) pricing schemes, and the price we use in the single-price setting can be drawn from one of the prices in the multi-price setting. Finally, in Subsect. 3.4 we consider an extension that does not assume independence between the job length and the value per time step.

3.1 Warm-Up: Uniform Distribution

As a warm-up example, assume that at any time step a job with length 1 or 2 appears with probability 50% each. The value per time step of a job is drawn from the uniform distribution over [0, 1]. Suppose that we set a price per time step p_1 for jobs of length 1 and p_2 for jobs of length 2.

Consider an arbitrary time step when the server is free. If the job drawn at that time step has length 1, then with probability p_1 it has value below p_1 and is rejected. In this case, the server passes one time step without a job. Otherwise, the job has value at least p_1 and is accepted. In this case, the expected welfare from executing the job is $\frac{1+p_1}{2}$. Similarly, if the job has length 2, then with probability p_2 it is rejected, and with probability $1-p_2$ it is accepted and yields expected welfare $2 \cdot \frac{1+p_2}{2} = 1 + p_2$ over two time steps. Letting c_w denote the expected welfare per time step assuming that the server is free at the current time step, we have

$$0 = \frac{1}{2}\left(-p_1c_w + (1-p_1)\left(\frac{1+p_1}{2} - c_w\right)\right) + \frac{1}{2}\left(-p_2c_w + (1-p_2)\left(1+p_2-2c_w\right)\right).$$

The two terms on the right hand side correspond to jobs of length 1 and 2, which are drawn with probability $\frac{1}{2}$ each. In the case that a job of length 2 is drawn, with probability p_2 it is rejected and the server is idle for one time step, during which it would otherwise have produced expected welfare c_w. With the remaining probability $1-p_2$ the job is accepted, yielding expected welfare $1+p_2$ over two time steps, during which the server would otherwise have produced expected welfare $2c_w$. The derivation for the term corresponding to jobs of length 1 is similar. By equating the expected welfare with the variable denoting this quantity, we arrive at the equation above.

Solving for c_w, we get

$$c_w(p_1, p_2) = \frac{\frac{(1-p_1)(1+p_1)}{2} + (1-p_2)(1+p_2)}{3-p_2}.$$

To maximize $c_w(p_1, p_2)$ over all values of p_1, p_2, we should set $p_1 = 0$. (Indeed, to maximize welfare we should always accept jobs of length 1 since they do not interfere with future jobs.) Then the value of p_2 that maximizes $c_w(p_1, p_2)$ is $p_2 = 3 - \sqrt{\frac{15}{2}} \approx 0.261$, yielding $c_w(p_1, p_2) = 6 - \sqrt{30} \approx 0.522$.

On the other hand, if we set the same price $p = p_1 = p_2$ for jobs with different lengths, our welfare per time step becomes

$$c_w(p) = \frac{\frac{(1-p)(1+p)}{2} + (1-p)(1+p)}{3-p} = \frac{3(1-p)(1+p)}{2(3-p)}.$$

This is maximized at $p = 3 - 2\sqrt{2} \approx 0.172$, yielding $c_w(p) = 9 - 6\sqrt{2} \approx 0.515$. Moreover, if we use either of the prices in the optimal price combination for the two-price setting as the single price, we get $c_w(0) = 0.5$ and $c_w\left(3 - \sqrt{\frac{15}{2}}\right) \approx 0.510$.

Next, we repeat the same exercise for revenue. We can derive the equations in the same way, with the only difference being that the revenue from accepting a job at price p is simply p. Letting c_r denote the revenue per time step, we have

$$0 = \frac{1}{2}\left(-p_1 c_r + (1-p_1)(p_1 - c_r)\right) + \frac{1}{2}\left(-p_2 c_r + (1-p_2)(2p_2 - 2c_r)\right).$$

Solving for c_r, we get

$$c_r(p_1, p_2) = \frac{(1-p_1)p_1 + 2(1-p_2)p_2}{3-p_2}.$$

To maximize c_r over all values of p_1, p_2, we should set $p_1 = 0.5$. (Indeed, to maximize revenue we should always set the monopoly price for jobs of length 1 since they do not interfere with future jobs.) Then the value of p_2 that maximizes $c_r(p_1, p_2)$ is $p_2 = 3 - \sqrt{\frac{47}{8}} \approx 0.576$, yielding $c_r(p_1, p_2) = 10 - \sqrt{94} \approx 0.304$.

On the other hand, if we set the same price $p = p_1 = p_2$ for jobs with different lengths, our revenue per time step becomes

$$c_r(p) = \frac{(1-p)p + 2(1-p)p}{3-p} = \frac{3(1-p)p}{3-p}.$$

This is maximized at $p = 3 - \sqrt{6} \approx 0.551$, yielding $c_r(p) = 15 - 6\sqrt{6} \approx 0.303$. Moreover, if we use either of the prices in the optimal price combination for the two-price setting as the single price, we get $c_r(0.5) = 0.3$ and $c_r\left(3 - \sqrt{\frac{47}{8}}\right) \approx 0.302$.

Observe that for both welfare and revenue, the maximum in the one-price setting is not far from that in the two-price setting. In addition, in both cases at least one of the two prices in the optimal price combination for the two-price setting, when used alone as a single price, performs almost as well as the maximum in the two-price setting. In the remainder of this section, we will show that this is not a coincidence, but rather a phenomenon that occurs for any set of job lengths, any probability distribution over job lengths, and any probability distribution over job values.

3.2 General 50% Approximation

In this subsection, we consider a general setting with an arbitrary number of job lengths. We show that even at this level of generality, it is always possible to obtain 50% of the welfare and revenue of setting an individual price for each job length by setting just one price. Although the optimal price in the one-price setting might be different from any of the prices in the multiple-price setting, we show that at least one of the prices in the latter setting can be used alone to achieve the 50% guarantee.

Assume that there are jobs of lengths $a_1 \leq a_2 \leq \cdots \leq a_n$ which appear at each time step with probability $r_1, r_2, \ldots, r_n$, respectively. Suppose that we set a price per time step p_i for jobs of length a_i. Recall that the value per time step of a job is drawn from a distribution with cumulative distribution function F and probability density function f.

The following lemma gives the formulas for the expected welfare and revenue per time step.

Lemma 3.1. *Let $S = a_1r_1 + \cdots + a_nr_n$ and $R = r_1 + \cdots + r_n$, and let c_w and c_r denote the expected welfare and revenue per time step, respectively. We have*

$$c_w(p_1, \ldots, p_n) = \frac{a_1r_1 \int_{x \geq p_1} \ell d\mu + \cdots + a_nr_n \int_{x \geq p_n} \ell d\mu}{S - ((a_1 - 1)r_1F(p_1) + \cdots + (a_n - 1)r_nF(p_n)) + (1 - R)} \tag{1}$$

and

$$c_r(p_1, \ldots, p_n) = \frac{a_1r_1(1 - F(p_1))p_1 + \cdots + a_nr_n(1 - F(p_n))p_n}{S - ((a_1 - 1)r_1F(p_1) + \cdots + (a_n - 1)r_nF(p_n)) + (1 - R)}. \tag{2}$$

In particular, if $p_1 = \cdots = p_n = p$, then $c_w(p) = \frac{S \int_{x \geq p} \ell d\mu}{S-(S-R)F(p)+(1-R)}$ and $c_r(p) = \frac{S(1-F(p))p}{S-(S-R)F(p)+(1-R)}$.

With the formulas for welfare and revenue in hand, we are ready to show the main result of this section, which exhibits that the worst-case approximation ratio for welfare or revenue between the one-price setting and the multiple-price setting is at least 50%. As we will see later in Subsect. 3.3, this bound is in fact tight, and it remains tight even when there are only two job lengths. Note that the bound holds for any number of job lengths, any distribution over job lengths, and any distribution over job values.

Theorem 3.1. *For any prices $p_1, p_2, \dots, p_n$ that we set in the multiple-price setting, we can achieve a welfare (resp. revenue, or any convex combination of welfare and revenue) approximation of at least* 50% *in the one-price setting by using one of the prices p_i as the single price.*

To prove Theorem 3.1, we work with the ratio $\frac{\max(c_w(p_1),\dots,c_w(p_n))}{c_w(p_1,\dots,p_n)}$ and show that it is at least $\frac{1}{2}$ for any $p_1, \dots, p_n$ (and similarly for revenue or any convex combination of welfare and revenue). Using the formula (1) for c_w given in Lemma 3.1, we can write the ratio in terms of the variables $A_i = \frac{\int_{x \ge p_i} \ell d\mu}{\int_{x \ge p_1} \ell d\mu}$ and $B_i = F(p_i)$ for $1 \le i \le n$. For any fixed values of B_i, we then deduce the values of A_i that minimize the ratio of interest. Finally, we show that the remaining expression is always at least $1/2$ no matter the values of B_i.

3.3 Tighter Bounds for Specific Parameters

Assume in this subsection that there are jobs of two lengths $a < b$ which appear at each time step with probability r_1 and r_2, respectively, where $r_1 + r_2 \le 1$. Suppose that we set a price per time step p_1 for jobs of length a and p_2 for jobs of length b. Recall that the value per time step of a job is drawn from a distribution with cumulative distribution function F and probability density function f.

Our next result exhibits a tight approximation bound for any fixed setting of the job lengths and their distribution.

Theorem 3.2. *For any prices p_1 and p_2 that we set in the two-price setting, we can achieve a welfare (resp. revenue, or any convex combination of welfare and revenue) approximation of at least*

$$\rho(a, b, r_1, r_2) := \frac{(ar_1 + br_2)(ar_1 + 1 - r_1)}{a(a-1)r_1^2 + a(b-1)r_1 r_2 + ar_1 + br_2}$$

in the one-price setting by setting either p_1 or p_2 alone. Moreover, this bound is the best possible even if we are allowed to set a price different from p_1 or p_2 in the one-price setting.

To prove this theorem, we work with the expression in terms of $B_i = F(p_i)$ that we have from the proof of Theorem 3.1. We then show that the expression is minimized when we take $B_1 = 0$ and $B_2 = 1$, meaning that the distribution on job values is bimodal. The proof method readily yields an example showing that our bound is tight, where the bimodal distribution on job values puts a large probability on a low value and a small probability on a high value.

If we fix the probabilities r_1, r_2, we can derive a tight worst-case bound over all possible job lengths a, b.

Theorem 3.3. *For fixed r_1, r_2, we have $\rho(a, b, r_1, r_2) \geq \frac{1}{1+r_1}$ for arbitrary a, b. Moreover, this bound is the best possible.*

Note that the fact that the bound is tight at $a = 1$ and $b \to \infty$ is consistent with the intuition that the further apart the job lengths are, the more welfare and revenue there is to be gained by setting different prices for the job lengths, and hence the worse the approximation ratio.

Finally, we show that we can obtain at least 50% of the welfare or revenue from setting two prices by using one of those prices.

Theorem 3.4. *For arbitrary a, b, r_1, r_2, we have $\rho(a, b, r_1, r_2) \geq \frac{1}{2}$. Moreover, this bound is the best possible.*

While we do not have a general formula for the worst-case approximation ratio for each choice of the parameters $a_1, \ldots, a_n, r_1, \ldots, r_n$ as we do for the case of two job lengths, the function h in the proof of Theorem 3.1 still allows us to derive a tighter bound for each specific case. Note that to find the minimum of h, it suffices to check $B_i = 0$ or 1 (see the full version of this paper [23] for details), so we only have a finite number of cases to check.

3.4 Extension

In this subsection, we show that by using a single price, we can obtain 50% of the welfare not only compared to using multiple prices, but also compared to the offline optimal welfare.[3] In fact, we will also not need the assumption that the job length and the value per time step are independent. However, the result only works for particular prices rather than arbitrary ones, and we cannot obtain tighter results for specific parameters using this method.

Theorem 3.5. *Assume that the job length and the value per time step are not necessarily independent. There exists a price p such that we can achieve a* 50% *approximation of the offline optimal welfare by using p as the single price.*

4 Multiple Servers

In this section, we assume that there are multiple servers, each of which receives jobs of various lengths. Under the assumption that the servers have the same probability of receiving no job at a time step, we show in Subsect. 4.1 an approximation bound of the welfare and revenue of setting one price for all servers compared to setting an individual price for each server. This yields a strong bound when at least one of the dimensions of the parameters is not too extreme, e.g., the number of servers or the job lengths are not too large. In Subsect. 4.2, we combine the newly obtained results with those from Sect. 3. Using a composition technique, we derive a general result that compares the welfare and revenue

[3] For the offline optimal welfare, we compute the limit of the expected average offline optimal welfare per time step as the time horizon grows.

obtained by a restricted mechanism that sets the same price for all servers and all job lengths against those obtained by a mechanism that can set a different price for each job length of each particular server. We show that even with the heavy restrictions, the former mechanism still provides a reasonable approximation to the latter one in a wide range of situations. Using similar techniques, we also obtain approximation bounds when this assumption does not hold but there is only one job length across all servers. The analysis of the latter setting can be found in the full version of this paper [23].

As in Sect. 3, our approximation results hold for arbitrary (i.e., not necessarily optimal) pricing schemes, and the price we use in the single-price setting can be drawn from one of the prices in the multi-price setting.

4.1 One Price per Server

Assume that at each time step, either zero or one job appears for each server $1 \leq j \leq n$. Server j receives jobs of length $a_{j1} \leq a_{j2} \leq \cdots \leq a_{jn_j}$ with probability $r_{j1}, r_{j2}, \ldots, r_{jn_j}$, respectively. Suppose that we set a price per time step p_j for all jobs on server j. Recall that the value per time step of a job is drawn from a distribution with cumulative distribution function F and probability density function f, and that we assume that $\sum_{i=1}^{n_j} r_{ji}$ is constant. Let $S_j = a_{j1}r_{j1} + \cdots + a_{jn_j}r_{jn_j}$ and $R = r_{j1} + \cdots + r_{jn_j}$.

Using the formula (1) for c_w given in Lemma 3.1, we find that the welfare per time step is

$$d_w(p_1, p_2, \ldots, p_n) = \sum_{j=1}^{n} \frac{\int_{x \geq p_j} \ell d\mu}{1 - \left(1 - \frac{R}{S_j}\right) F(p_j) + \frac{1-R}{S_j}}.$$

If we set the same price $p = p_1 = \cdots = p_n$ for different servers, our welfare per time step becomes $d_w(p) = \sum_{j=1}^{n} \frac{\int_{x \geq p} \ell d\mu}{1-\left(1-\frac{R}{S_j}\right)F(p)+\frac{1-R}{S_j}}$. The formulas $d_r(p_1, p_2, \ldots, p_n)$ and $d_r(p)$ for revenue are similar but with the terms $\int_{x \geq p_j} \ell d\mu$ replaced by $(1 - F(p_j))p_j$.

We show that if at least one dimension of the parameters is not too extreme, e.g., the number of servers or the job lengths are bounded, then we can obtain a reasonable approximation of the welfare and revenue in the multi-price setting by setting just one price.

Theorem 4.1. *For any prices $p_1, p_2, \ldots, p_n$ that we set in the multiple-price setting, we can achieve a welfare (resp. revenue, or any convex combination of welfare and revenue) approximation of at least*

$$\max\left(\frac{1}{H_n}, \frac{M-1}{M \ln M}\right)$$

in the one-price setting, where $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n$ is the nth Harmonic number and $M = \max_{i,j} \frac{S_i}{S_j}$.

In particular, if all job lengths are bounded above by c, then $R \leq S_j \leq cR$ for all $1 \leq j \leq n$, and so $\max_{i,j} \frac{S_i}{S_j} \leq c$. The theorem then implies that the approximation ratio is at least $\frac{c-1}{c \ln c}$.

4.2 Multiple Prices per Server

Assume as in Subsect. 4.1 that at each time step, server j receives jobs of length $a_{j1} \leq a_{j2} \leq \cdots \leq a_{jn_j}$ with probability $r_{j1}, r_{j2}, \ldots, r_{jn_j}$, respectively. In this subsection, we consider setting an individual price not only for each server but also for each job length of that server. In particular, suppose that we set a price per time step p_{ji} for jobs of length a_{ji} on server j. Recall that the value per time step of a job is drawn from a distribution with cumulative distribution function F and probability density function f, and that we assume that $\sum_{i=1}^{n_j} r_{ji}$ is constant. Let $S_j = a_{j1}r_{j1} + \cdots + a_{jn_j}r_{jn_j}$.

We will compare a setting where we have considerable freedom with our pricing scheme and can set a different price p_{ji} for each job length a_{ji} on each server j with a setting where we have limited freedom and must set the same price p for all job lengths and all servers. We show that by "composing" our results on the two dimensions, we can obtain an approximation of the welfare and revenue of setting different prices by setting a single price.

Theorem 4.2. *For any prices p_{ji}, where $1 \leq j \leq n$ and $1 \leq i \leq n_j$ for each j, that we set in the multiple-price setting, we can achieve a welfare (resp. revenue, or any convex combination of welfare and revenue) approximation of at least*

$$\frac{1}{2} \cdot \max\left(\frac{1}{H_n}, \frac{M-1}{M \ln M}\right)$$

in the one-price setting, where $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n$ is the nth Harmonic number and $M = \max_{i,j} \frac{S_i}{S_j}$.

If we have tighter approximations for either the "different prices for different job lengths" or the "different prices for different servers" dimension, for instance by knowing the values of some of the parameters, then the same composition argument yields a correspondingly tighter bound.

5 Conclusion

In this paper, we study how well simple pricing schemes that are oblivious to certain parameters can approximate optimal schemes with respect to welfare and revenue, and prove several results when the simple schemes are restricted to setting the same price for all servers or all job lengths. Our results provide an explanation of the efficacy of such schemes in practice, including the one shown in Fig. 1 for virtual machines on Microsoft Azure. Since simple schemes do not require agents to spend time and resources to determine their specific parameter values, our results also serve as an argument in favor of using these schemes in a range of applications. It is worth noting that as all of our results are of worst case nature, we can expect the guarantees on welfare and revenue to be significantly better than these pessimistic bounds in practical instances where the parameters are not adversarially tailored.

We believe that there is still much interesting work to be done in the study of simple pricing schemes for the cloud. We conclude our paper by listing some intriguing future directions.

- In many scheduling applications, a job can be scheduled online to any server that is not occupied at the time. Does a good welfare or revenue approximation hold in such a model?
- Can our results be extended to models with more fluid job arrivals, for example one where several jobs can arrive at each time step?
- Can we approximate welfare and revenue simultaneously? A trivial randomized approach would be to choose with equal probability whether to approximate welfare or revenue. According to Theorem 3.1, this yields a 1/4-approximation for both expected welfare and expected revenue of the single-price setting in comparison to the multi-price setting for job lengths.

References

1. Abhishek, V., Kash, I.A., Key, P.: Fixed and market pricing for cloud services. In: The 7th Workshop on the Economics of Networks, Systems and Computation (2012)
2. Amazon EC2 Spot Instances Pricing (2017). http://aws.amazon.com/ec2/spot/pricing. Accessed 1 Aug 2017
3. Azar, Y., Kalp-Shaltiel, I., Lucier, B., Menache, I., Naor, J.S., Yaniv, J.: Truthful online scheduling with commitments. In: Proceedings of the Sixteenth ACM Conference on Economics and Computation, pp. 715–732 (2015)
4. Microsoft Azure Pricing Calculator (2016). http://azure.microsoft.com/en-us/pricing/calculator. Accessed 19 Sept 2016
5. Babaioff, M., Blumrosen, L., Dughmi, S., Singer, Y.: Posting prices with unknown distributions. In: Innovations in Computer Science - ICS 2010, pp. 166–178 (2011)
6. Blumrosen, L., Holenstein, T.: Posted prices vs. negotiations: an asymptotic analysis. In: Proceedings of the 9th ACM Conference on Electronic Commerce, p. 49 (2008)
7. Chawla, S., Hartline, J.D., Malec, D.L., Sivan, B.: Multi-parameter mechanism design and sequential posted pricing. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, pp. 311–320 (2010)
8. Cohen, I.R., Eden, A., Fiat, A., Jez, L.: Pricing online decisions: beyond auctions. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 73–91 (2015)
9. Cohen-Addad, V., Eden, A., Feldman, M., Fiat, A.: The invisible hand of dynamic market pricing. In: Proceedings of the 2016 ACM Conference on Economics and Computation, pp. 383–400 (2016)

10. Columbus, L.: Roundup of cloud computing forecasts and market estimates, 2016 (2016). http://www.forbes.com/sites/louiscolumbus/2016/03/13/roundup-of-cloud-computing-forecasts-and-market-estimates-2016. Accessed 19 Sept 2016
11. Dehghani, S., Kash, I.A., Key, P.: Online stochastic scheduling and pricing the cloud. Working Paper (2016)
12. Dierks, L., Seuken, S.: Cloud pricing: the spot market strikes back. In: The Workshop on Economics of Cloud Computing (2016)
13. Disser, Y., Fearnley, J., Gairing, M., Göbel, O., Klimm, M., Schmand, D., Skopalik, A., Tönnis, A.: Hiring secretaries over time: the benefit of concurrent employment. CoRR, abs/1604.08125 (2016)
14. Dütting, P., Fischer, F.A., Klimm, M.: Revenue gaps for discriminatory and anonymous sequential posted pricing. CoRR, abs/1607.07105 (2016)
15. Ezra, T., Feldman, M., Roughgarden, T., Suksompong, W.: Pricing identical items. CoRR, abs/1705.06623 (2017)
16. Feldman, M., Gravin, N., Lucier, B.: Combinatorial auctions via posted prices. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 123–135 (2015)
17. Friedman, E.J., Ghodsi, A., Psomas, C.: Strategyproof allocation of discrete jobs on multiple machines. In: Proceedings of the Fifteenth ACM Conference on Economics and Computation, pp. 529–546 (2014)
18. Friedman, E., Rácz, M.Z., Shenker, S.: Dynamic budget-constrained pricing in the cloud. In: Barbosa, D., Milios, E. (eds.) CANADIAN AI 2015. LNCS, vol. 9091, pp. 114–121. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18356-5_10
19. Hoy, D., Immorlica, N., Lucier, B.: On-demand or spot? Selling the cloud to risk-averse customers. In: Cai, Y., Vetta, A. (eds.) WINE 2016. LNCS, vol. 10123, pp. 73–86. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-54110-4_6
20. Jain, N., Menache, I., Naor, J.S., Yaniv, J.: A truthful mechanism for value-based scheduling in cloud computing. In: Persiano, G. (ed.) SAGT 2011. LNCS, vol. 6982, pp. 178–189. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24829-0_17
21. Jain, N., Menache, I., Naor, J.S., Yaniv, J.: Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters. In: Proceedings of the 24th ACM Symposium on Parallelism in Algorithms and Architectures, pp. 255–266 (2012)
22. Kash, I.A., Key, P.: Pricing the cloud. IEEE Internet Comput. **20**(1), 36–43 (2016)
23. Kash, I.A., Key, P., Suksompong, W.: Simple pricing schemes for the cloud. CoRR, abs/1705.08563 (2017)
24. Lucier, B., Menache, I., Naor, J.S., Yaniv, J.: Efficient online scheduling for deadline-sensitive jobs. In: Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures, pp. 305–314 (2013)
25. Wang, C., Ma, W., Qin, T., Chen, X., Hu, X., Liu, T.-Y.: Selling reserved instances in cloud computing. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 224–230 (2015)
26. Zhang, H., Li, B., Jiang, H., Liu, F., Vasilakos, A.V., Liu, J.: A framework for truthful online auctions in cloud computing with heterogeneous user demands. In: Proceedings of the IEEE INFOCOM 2013, pp. 1510–1518 (2013)