Recouping Energy Costs from Cloud Tenants: Tenant Demand Response Aware Pricing Design

Cheng Wang, Neda Nasiriani, George Kesidis, Bhuvan Urgaonkar, Qian Wang,
Lydia Y. Chen†, Aayush Gupta‡, Robert Birke†
The Pennsylvania State University, †IBM Research Zurich, ‡IBM Research Almaden

ABSTRACT

As energy costs become increasingly greater contributors to a cloud provider's overall costs, it is important for the cloud to recoup these energy costs from its tenants for profitability via appropriate pricing design. The poor predictability of real-world tenants' demand and demand responses (DRs) make such pricing design a challenging problem. We formulate a leader-follower game-based cloud pricing framework with the goal of maximizing cloud's profit. The key distinguishing aspect of our approach is our emphasis on modeling both the cloud and its tenants as working with low predictability in their inputs. Consequently, we model them as employing myopic control with short-term predictive models. Our empirical evaluation using tenant trace from IBM production data centers shows that (i) cloud's profit and VM prices are sensitive to the tradeoffs between its energy costs, tenant's demand and DR, and (ii) the cloud's estimation of tenants' demands/DR may significantly affect its profitability.

1. INTRODUCTION

The electric utility bills of data centers make up significant portions of their overall expenses and are fast approaching the capital expenditure towards IT infrastructure itself. Studies from large cloud/IT providers such as Google [2] and Amazon [12] show that the electric utility bill amounts to 10-20% of the overall costs of their state-of-the-art data centers. More alarmingly, perhaps, it is likely that these energy bills will become even larger contributors to data center costs as energy prices increase in the future [27]. Consequently, the pricing mechanism employed by a cloud provider (or simply "cloud" henceforth) to recoup these energy costs from its customers (i.e., "tenants") has important implications for its profitability and has recently emerged as a topic of much interest [22, 18, 31].

Pricing design for a cloud is made challenging by uncertainty in tenant workloads as well as in electric utility prices. Although much related work in cloud pricing design makes assumptions of predictability in utility prices and tenant workloads for theoretic tractability (see discussions in Sections 2 and 5 for salient examples), many real-world prices and workloads exhibit poor predictability and are best considered non-stationary. Even in cases where utility prices or workloads can be reasonably predicted via higher-order predictors [9, 5], the resulting control problems (including pricing design) are likely to be computationally difficult to solve.

We envision that a whole new source of complexity in pricing design will arise in the near future due to the emergence (or increasing adoption) of *price sensitive tenant behavior*. Demand response (DR) has been recently identified as being important for the profitable operation of data centers [29], and we foresee even individual tenants (perhaps starting with large and sophisticated ones) similarly carrying out DR of their own. We use the phrase "tenant DR" (or simply DR when the meaning is clear) to describe strategic resource procurement by tenants in response to prices set by the cloud. Many enterprises and businesses are moving increasing portions of their IT needs to various cloud providers, a trend that is likely to continue in the foreseeable future [11]. As these tenants become increasingly invested into cloud computing, it is reasonable to expect that price sensitive DR will become increasingly important for their profitability. As a salient example, the video steaming giant Netflix already procures all of its vast computational needs from Amazon's EC2 cloud and employs certain forms of DR [21]. Generally speaking, *tenant DR will be a priori unknown to the cloud provider*, making the problem of price determination for the cloud even more complicated.

Using real-world data for tenant workloads and utility prices, we study the problem of *virtual machine (VM) pricing design for a cloud whose tenants engage in price-sensitive demand response*. Our approach involves modeling this ecosystem via a leader-follower game with the cloud being the leader and tenants the followers. Whereas some existing work has investigated cloud's pricing design assuming that tenants' workloads and DRs can be well predicted/inferred (see Section 5), a key feature of our approach is our emphasis on modeling various participants as working with low predictability. Specifically, we assume that the cloud and the tenants find long-term prediction difficult and instead choose to work with relatively short-term predictions (of cloud prices and workloads for tenants and of utility prices and tenant workloads/DRs for the cloud). In other

words, the cloud and the tenants employ "myopic" control approaches with objectives of maximizing only their respective short-term profits.

Our contributions are along both analytical and empirical lines. On the analytical front, we formulate a leader-follower game-based cloud pricing framework with the goal of maximizing cloud's profit. In this model, the cloud employs short-term sequential decision making (VM pricing) with prediction models for estimating tenants' DRs. To the best of our knowledge, this is novel in the area of cloud pricing. Our most significant contributions are the following key findings of the empirical evaluation based on real-world utility prices and tenant demand workloads from production data centers run by IBM: (i) cloud's profit and VM prices are sensitive to the tradeoffs between its energy costs, tenant's demand and DR, and (ii) the cloud's estimation of tenants' demands/DR may significantly affect its profitability. Our analysis based on the cloud and the tenants carrying out decision-making restricted by short-term predictability yields non-trivial and surprising findings compared to those offered by existing work that simplifies input complexity for tractable analytical solutions. For example, optimal *long term* profitability is not achieved by the necessarily "myopic" framework (both in terms of revenue objectives and estimation techniques), even under otherwise ideal circumstances such as perfect forecasting of tenants' DRs.

2. BACKGROUND

2.1 System model and assumptions

A cloud provider procures various resources from different kinds of utility providers and/or renders and constructs virtualized IT resources from these for its tenants. E.g., a cloud provider might purchase electric power from an electric utility company, network bandwidth from an Internet Service Provider (ISP), and servers and storage once every few years from IT infrastructure retailers. The cloud provider then creates a variety of resources for its tenants such as virtual machines (VMs), storage, software services, etc., with a variety of pricing options such as on-demand pricing, reservation-based pricing, spot pricing (with tenants' bidding) [1].

We consider a system model that simplifies the above diversity significantly by considering a single resource - energy - procured by the cloud, and a single resource - a single type of VM - sold to the tenants. Figure 1 illustrates our model and helps compare it with a more general ecosystem. In the following, we describe key simplifying assumptions in our model.

Utility-Cloud: We assume that the cloud is charged by the electric utility company based on a time-varying electricity pricing scheme. Any DR the cloud does is accomplished *implicitly* via the behavior of tenants (as described next) incentivized by the VM prices set by the cloud. In future work we will also incorporate explicit DR by the cloud (in addition to and/or complementary to that carried out by the tenants). We further assume that the utility prices are unaffected by the power demands posed by the cloud. The goal of the cloud is to maximize its own profit by modulating VM prices based on its short-term predictions of tenants' demands/DRs.

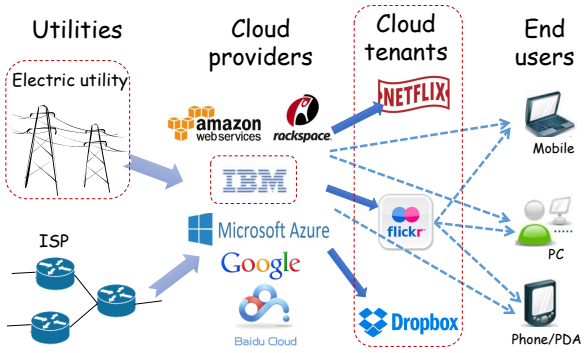


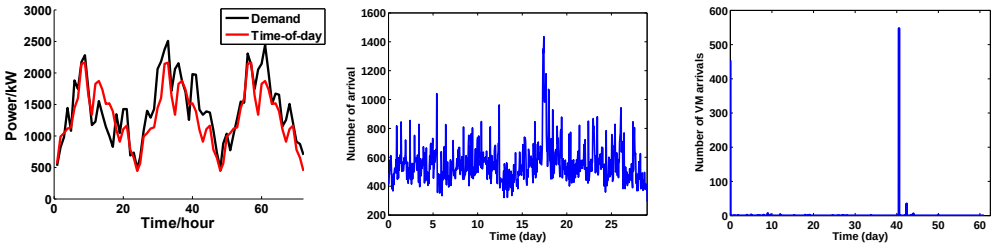
Figure 1: Illustration of a general cloud ecosystem. We highlight the elements we focus on in our simplified model.

Cloud-Tenant: The cloud charges its tenants using a time-varying VM pricing scheme. We assume VMs to have a fixed resource capacity. In practice VM resource capacity could be dynamic due to cloud's resource management actions such as VM consolidation, under-provisioning, etc. E.g., Netflix employs certain forms of control in response to such variations in the resource capacity of Amazon EC2 VMs [21]. Tenants are assumed to be price sensitive and carry out DR with the goal of maximizing their respective net utility. We assume that a tenant's DR is carried out by "delaying" its workload, i.e., by deferring (or suspending) VMs with a delay penalty that captures the tenant's revenue loss/performance degradation due to DR. Other forms of DR such as strategic VM procurement and workload consolidation, etc., are interesting future work.

2.2 Motivating myopic control based on short-term predictions

Plenty of evidence from real-world data centers shows that their workloads (and correspondingly power demands) can be very complex [24, 16, 5, 15]. In particular, data center workloads exhibit varying degrees of predictability. For example, the Facebook demand shown in Figure 2(a) lends itself to low complexity predictors, e.g., by subtracting the time-of-day effects from raw demand and modeling the residual demand as Markovian. On the other hand, job arrivals at a Google cluster (Figure 2(b)) exhibit poorer predictability: upon plotting the autocorrelation of the residue after detrending for time-of-day effects from Figure 2(b) (not shown here due to lack of space), we find that rather high-order predictors might be required that would result in computationally complex control formulations. As an example with even poorer predictability, in Figure 2(c) we observe an unexpected burst of VM arrivals on the 40-th day from the VM demand trace of a tenant belonging to an IBM production data center. Most existing work on cloud pricing design is suitable only for workloads like in Figure 2(a). In this work, we choose to focus on scenarios exemplified by Figures 2(b) and (c).

A whole new source of complexity arises in our problem domain due to the tenants' DRs and the cloud's lack of knowledge thereof. Different tenants could have different sensitivities to price changes. As an example, tenants that run performance-centric applications, e.g., Web services, might want guaranteed performance even at a high price (i.e., less price-sensitive) whereas some batch jobs could be scheduled



(a) Power demand from a facebook data center [6].

(b) Job arrivals at a Google cluster [10].

(c) VM arrivals for an IBM tenant.

Figure 2: Examples of real-world data center workloads with different degrees of predictability.

with more flexibility (i.e., more price-sensitive). Furthermore, tenants' DRs could also be time-varying and/or depend on the tenant's own customers' demand. For example, when streaming a very popular event, a streaming video server tenant hosted on the cloud might be willing to procure more computing resources and be willing to pay a higher price to the cloud to satisfy the demands of the large number of clients with guaranteed QoS than when it streams a less popular event.

In the face of these complexities, optimal control techniques that rely upon effective predictors for inputs (e.g., based on Markov Decision Processes) might not work well. Therefore, we choose to work with "myopic" control for both the cloud and the tenants wherein (a) the cloud only maximizes its short-term profit and sets VM prices based on short-term prediction/estimation of tenant's VM demand and DR, and (b) the tenant also only optimizes its short-term net utility and defers VMs myopically by only considering the penalty (i.e., revenue loss) of deferring VMs in the short term.

An Important Warning. An implication of the complexity of workloads and DRs in our problem domain is that we will be unable to compare the quality of our solutions against an offline optimal (due to the computational complexity of such a formulation). Instead, our "baseline" would be a solution that is "optimal" only with respect to our myopically defined objective (i.e., short-term profit maximization) which does not necessarily maximize the long-term profit. It will be important to keep in mind that we use "optimal" in this sense when we present our empirical evaluation in Section 4.

Cloud's estimation of tenants' DR. Due to the aforementioned complexity in tenants' DRs, we assume that the cloud only estimates the tenants' **aggregate DR** for computational tractability. Extensions to consider different types of tenants whose behavior needs to be predicted separately form future work. We consider a time-slotted system wherein the cloud needs to determine the VM price θ_{t^-} at the beginning of time-slot t. Here t^- implies that θ_{t^-} is determined by the cloud *before* the actual VM demand during time-slot t is revealed by the tenants. Having clarified this causal order, we simply use θ_t instead of θ_{t^-} for notational simplicity in the following.

Denoting as x_t the aggregate VM demands from all tenants during time-slot t, the cloud employs predictive models to forecast x_t. As an example, if it uses a first-order auto-regressive estimator with exogenous input (ARX), i.e., $\hat{x}_t = A_t x_{t-1} + B_t \theta_t$ where A_t, B_t are model parameters that are updated online recursively by using historical data. The rationale behind the above prediction model is that (i) the tenants' aggregate admitted VM demand may depend significantly on that of the previous time-slot, which is reflected by the AR term $A_t x_{t-1}$, and (ii) VM demand is likely to decrease as price increases, which is captured by $B_t \theta_t$ with $B_t < 0$.

To see how the cloud's profitability might be affected by its estimation of tenant's aggregate DR, we consider several different predictors whose key properties are shown in Table 1. We report experiment results with all these predictors in Section 4.

Name	Predictor	DoF
ARX	$\hat{x}_t = A_t x_{t-1} + B_t \theta_t$	2
Affine	$\hat{x}_t = B_{1,t}\theta_t + B_{0,t}$	2
Quad (quadratic)	$\hat{x}_t = B_{2,t}\theta_t^2 + B_{1,t}\theta_t + B_{0,t}$	3
PW (piecewise linear)	$\hat{x}_t = \sum_{i=1}^{M}(B_{i,t}\theta_t + C_{i,t})$	$2M$

Table 1: Predictors for cloud's estimation of tenants' aggregate DR. "DoF": degrees of freedom.

Generally speaking, the cloud could be tempted to use more complex predictors instead of the prosaic predictors aforementioned. However, higher order predictors would rely on larger training sets to tune their parameters (and sometimes even hyper-parameters) adaptively. When the real-world tenants' DR models are (highly) time-varying, the cloud might not be able to find sufficiently informative prior training data for system identification (since outdated data might not be helpful). Furthermore, the real-world tenant's demand we consider has poor predictability, which further adds to the difficulty of applying complex high-order predictors. Therefore, we rely on the above parsimonious predictors in our design.

3. GAME-THEORETIC PRICING DESIGN

In this section, we design a leader/follower game-based cloud pricing framework with the goal of maximizing the cloud's profit. Figure 3 illustrates the sequential decision making process of our game. As shown, at the beginning of time-slot t, there are two steps: in step I (cloud's move), assuming that the cloud has perfect knowledge (or is reasonably well informed) of the future electricity prices in K future slots $\alpha_t, \alpha_{t+1}, ..., \alpha_{t+K}$, it determines VM prices $\theta_t, \theta_{t+1}, ..., \theta_{t+K}$ to maximize its profit during these $(K+1)$ slots using predictions of tenants' aggregate DR. Towards this, the cloud forecasts the tenants' aggregate VM demands for the next $K+1$ time-slots under the VM prices it sets. Since its prediction of tenant's demand will likely be increasingly less

accurate as the cloud forecasts farther into the future, it only adopts θ_t as the VM price for time-slot t. Estimates of $\theta_{t+1}, ..., \theta_{t+K}$ are either discarded or sent to the tenants as guiding pricing signals, denoted as $\theta'_{t+1}, ..., \theta'_{t+K}$ to distinguish them from the prices actually used.

Step II (tenant's move) has two sub-steps: In step II.a, the tenants decide how many VMs to admit and how many VMs to defer based on their respective demands, delay costs, and cloud's VM price θ_t and pricing signals $\theta'_{t+1}, ..., \theta'_{t+K}$. In step II.b, the tenants choose which VMs to defer to future time-slots according to priority, deadline or any specific performance requirement (if any) of their VMs.

3.1 Step I: Cloud's control

We develop our formulation for $K = 1$, i.e., the cloud maximizes its profit over two consecutive time-slots, and note that it can be easily generalized to cases with larger K. The cloud would predict its revenue during time-slots t and $t+1$ as: $\theta_t \hat{x}_t + \theta'_{t+1} \hat{x}_{t+1}$.

Assuming that the fraction of cloud's overall costs that comes from its electricity bill is c ($c < 1$), and that the energy consumed by $\hat{x}_t$ VMs is $g(\hat{x}_t)$[1], the **cloud's cost** during time-slots t and $t+1$ can be written as: $\frac{1}{c}\{\alpha_t g(\hat{x}_t) + \alpha_{t+1} g(\hat{x}_{t+1})\}$.

At the beginning of time-slot t, the cloud takes the electricity prices α_t and α_{t+1} as inputs, and outputs the actual VM price θ_t and the estimated pricing signal θ'_{t+1} by solving the following profit maximization problem:

$$\max_{\theta_t, \theta'_{t+1}} \quad (\theta_t \hat{x}_t + \theta'_{t+1} \hat{x}_{t+1}) - \frac{1}{c}\{\alpha_t g(\hat{x}_t) + \alpha_{t+1} g(\hat{x}_{t+1})\}$$

Subject to

$$0 \le \hat{x}_t = A_t x_{t-1} + B_t \theta_t$$
$$0 \le \hat{x}_{t+1} = A_t \hat{x}_t + B_t \theta'_{t+1}$$
$$\theta_t \hat{x}_t \ge \frac{1}{c} \alpha_t g(\hat{x}_t)$$
$$\theta'_{t+1} \hat{x}_{t+1} \ge \frac{1}{c} \alpha_{t+1} g(\hat{x}_{t+1})$$

where x_{t-1} and θ_{t-1} are known (to the cloud) at the beginning of time-slot t. The first two constraints represent the cloud's estimations of the tenants' aggregate demands during time-slots t and $t+1$ using an ARX model. The last two constraints imply that the cloud's profit should be non-negative. Whereas we show the formulation using an ARX predictor, it can be replaced by a different predictor (as we explored in Section 4).

3.2 Step II: Tenant's control

At the beginning of time-slot t, upon receiving θ_t and θ'_{t+1} from the cloud[2], tenant i has to determine the number of VMs to be admitted, denoted as $x_{i,t}$, and the number of VMs to be deferred (with certain delay cost) to the next time-slot, denoted as $\eta_{i,t}$, and choose from all VMs which to defer.

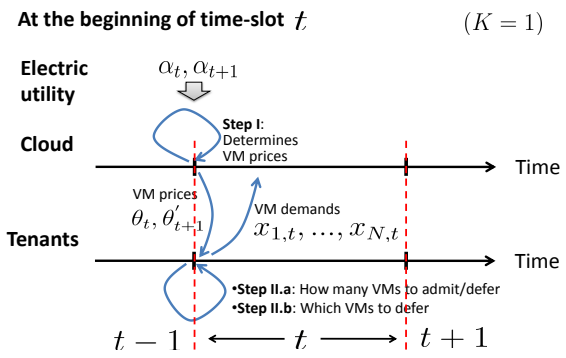


Figure 3: Illustration of our leader/follower game.

Step II.a: How many VMs to admit? (Determine the **control variable** $x_{i,t}$).

Each VM is characterized as a two-tuple: $\{a_{i,k}, l_{i,k}\}$ wherein $a_{i,k}$ denotes the arrival time (beginning of a time-slot) of tenant i's k-th VM, and $l_{i,k}$ is the lifetime of that VM. If we defer a VM from time-slot t to time-slot $t+1$, the lifetime of the VM will be increased such that $\tilde{l}_{i,k,t+1} = \tilde{l}_{i,k,t} + 1$, where $\tilde{l}_{i,k,t}$ denotes the lifetime of the VM at the beginning of time-slot t. Note that $l_{i,k} = \tilde{l}_{i,k,t}$ when $t = a_{i,k}$.

Define the **VM arrivals-minus-departures** at the beginning of time-slot t as follows:

$$\Delta x_{i,t} := \sum_k 1\{a_{i,k} = t\} - \sum_k 1\{a_{i,k} + \tilde{l}_{i,k,t} = t\}$$

If $x_{i,t}$ is the number of VMs that are active during time-slot t, and $X_{i,t}$ the total number of VMs in the system that are eligible for deferral/admittance at the beginning of time-slot t, then

$$X_{i,t} = x_{i,t-1} + \eta_{i,t-1} + \Delta x_{i,t}.$$

Since some of the VMs will continue to be active ($x_{i,t}$) while other VMs will be deferred to $t+1$ ($\eta_{i,t}$) at the beginning of time-slot t, we have

$$\eta_{i,t} = x_{i,t-1} + \eta_{i,t-1} + \Delta x_{i,t} - x_{i,t}$$

Figure 4 illustrates our model for tenant i's demand and control actions.

Define $f_i(.)$ as tenant i's utility function, which is assumed concave and increasing in number of VMs. Then we can express **tenant i's revenue** during time-slot t as $f_i(x_{i,t})$. Similarly, we can write **tenant i's costs** as $\theta_t x_{i,t} + \pi_{i,t} \eta_{i,t}$ where $\pi_{i,t}$ is the tenant-specific unit delay cost during time-slot t.

Now given θ_t and θ'_{t+1} from the cloud, tenant i can maximize its net profit over, e.g., two-consecutive time-slots, by

[1]In evaluation, we use $g(\hat{x}_t) = \delta \hat{x}_t$, which could be a reasonable model assuming fixed degree of VM consolidation (e.g., fixed number of VMs per physical machine).

[2]Note that it is also possible that the cloud exposes electricity prices α_t to the tenants for better forecasting and DR. Naturally, in this case, the tenant could attempt to forecast the cloud's VM prices in deciding whether and which VMs to defer.

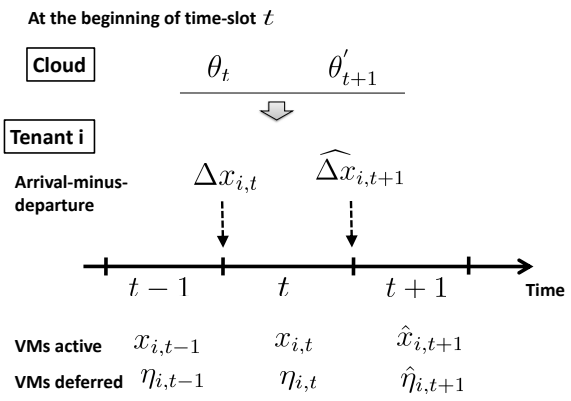


Figure 4: Illustration of tenant i's demand and control actions.

determining $x_{i,t}$ and $\hat{x}_{i,t+1}$ as follows:

$$\max_{x_{i,t},\,\hat{x}_{i,t+1}} \quad f_i(x_{i,t}) - (\theta_t x_{i,t} + \pi_{i,t}\eta_{i,t}) + f_i(\hat{x}_{i,t+1}) - (\theta'_{t+1}\hat{x}_{i,t+1} + \pi_{i,t+1}\hat{\eta}_{i,t+1})$$

Subject to

$$0 \le \eta_{i,t} = x_{i,t-1} + \eta_{i,t-1} + \Delta x_{i,t} - x_{i,t}$$
$$0 \le \hat{\eta}_{i,t+1} = x_{i,t} + \eta_{i,t} + \widehat{\Delta x}_{i,t+1} - \hat{x}_{i,t+1}$$

where $x_{i,t-1}, \Delta x_{i,t}, \eta_{i,t-1}$ are known at the beginning of t, and $\widehat{\Delta x}_{i,t+1}$ needs to be forecasted beforehand via predictive model, e.g., $\widehat{\Delta x}_{i,t+1} = \gamma_1 \Delta x_{i,t} + \gamma_2 \Delta x_{i,t-1}$. Here we adopt the idea of "receding horizon control" [19]: the tenant only implements $x_{i,t}$ and discards $\hat{x}_{i,t+1}$ assuming that its prediction (and control actions henceforth) can be further improved when time advances to time-slot $t+1$.

Step II.b: Which VMs to defer?

It is a design choice for the tenants to decide which VMs to defer. The tenant could cast a generic VM scheduling problem to minimize the average delay or deadline violation given information about the VMs' priority, deadline or specific performance requirement (if any). However, without the above information in our work, we have to defer VMs based on heuristics, e.g, deferring the VMs that have stayed in the system for least amount of time (assuming that those VMs might be short-lived and thus less important than others). That is, at the beginning of time-slot t, if $\eta_{i,t} = 1$, the tenant chooses VM k^* such that $k^* = \arg\min_k\{t - a_{i,k} \mid a_{i,k} \le t \le a_{i,k} + \tilde{l}_{i,k,t}\}$. Here $a_{i,k} \le t \le a_{i,k} + \tilde{l}_{i,k,t}$ implies that VM k enters the system before or at the beginning of t and does not leave at the end of t where $t - a_{i,k}$ is the number of time-slots that the VM has been in the system by time-slot t. If $\eta_{i,t} > 1$, then the tenant chooses the VMs that have been in the system for shortest amount of time by time-slot t.

Note that $\tilde{l}_{i,k,t}$ is generally not known at the beginning of time-slot t. However, for VM k, we do know the cumulative delay that it experiences at the beginning of t: $\tilde{l}_{i,k,t} - l_{i,k}$. Therefore, we can choose the VMs to defer based on $t - a_{i,k}$ or $\tilde{l}_{i,k,t} - l_{i,k}$ or alternatively a combination of them.

Remark 1. It is easy to check that the tenant's profit in time-slot $t+1$ does not depend on $x_{i,t}$ due to our assumption that the tenant's revenue is only a function of the number of VMs it admits in the current time-slot (and proportional to VM lifetime henceforth). Therefore, the above optimization problem reduces to the following "myopic" control that only maximizes tenant's profit over a single time-slot:

$$\max_{x_{i,t}} f_i(x_{i,t}) - (\theta_t x_{i,t} + \pi_{i,t}\eta_{i,t})$$

Subject to

$$0 \le \eta_{i,t} = x_{i,t-1} + \eta_{i,t-1} + \Delta x_{i,t} - x_{i,t}$$

Note that now the tenant does not need to know θ'_{t+1} in this myopic control. In addition, prediction of future demand is not needed since the tenant only maximize profit within the current time-slot. In evaluation, we will use this simplified myopic control for tenant's DR for simplicity.

Remark 2. In the real world, we can also have scenarios wherein the revenue is not proportional to the lifetime of the VM. In that case, one simple idea is to define the utility function $f_i(x_{i,t} + \beta\hat{x}_{i,t+1})$ for a parameter $0 < \beta < 1$. In this case, the tenant will have to take the guiding price signal θ'_{t+1} to maximize its profit. Intuitively this utility function implies that the tenant's revenue diminishes as the service time of a VM increases. One such example could be a Web search engine as a tenant, wherein the usefulness of additional search results (at the expense of larger response times) reduces [13]. Another example is a video streaming server wherein the extra benefit of increasing quality of video diminishes with bandwidth needs.

3.3 Understanding cloud-tenant interactions

Before carrying out our empirical evaluation, we try to get a preliminary understanding of cloud-tenant interactions in our model by making some simplifying assumptions about the players' knowledge of future inputs. Recall that all the "optimal solutions" discussed in this section are only optimal w.r.t. the "myopic" objective (short-term optimization horizon), which do not necessarily optimize the long-term profits.

The following claim reveals the tenant's optimal demand response behavior under our assumptions. Due to space limit, we present all the proofs in a technical report [28].

CLAIM 1. *Assuming that the tenant's utility function $f_i(\cdot)$ is strictly concave and non-decreasing, the tenant's control problem has the following closed-form optimal solution:*

$$x^*_{i,t} = \begin{cases} X_{i,t}, & \text{if } \theta_t \le \tilde{\theta}_t \\ (f'_i)^{-1}(\theta_t - \pi_{i,t}), & \text{otherwise} \end{cases}$$

where $X_{i,t} = x_{i,t-1} + \eta_{i,t-1} + \Delta x_{i,t}$, $(f'_i)^{-1}(\cdot)$ is the inverse function of the derivative $f'_i(\cdot)$, and $\tilde{\theta}_t$ is the solution of equation $(f'_i)^{-1}(\theta_t - \pi_{i,t}) = X_{i,t}$ with θ_t as variable[3].

In particular, if $f_i(x_{i,t}) = a\log(bx_{i,t}+1)$ with $a, b > 0$, then $x^*_{i,t} = \frac{a}{\theta_t - \pi_{i,t}} - \frac{1}{b}$. We show the tenant's optimal demand

[3]Note that $f'_i(\cdot)$ is monotonically decreasing since $f_i(\cdot)$ is concave and increasing. Thus there exists a unique solution for the equation $(f'_i)^{-1}(\theta_t - \pi_{i,t}) = X_{i,t}$.

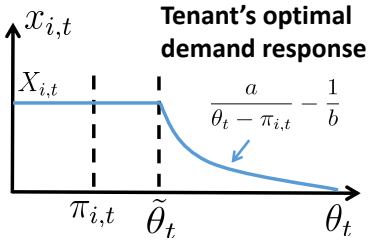


Figure 5: Illustration of tenant i's optimal demand response assuming $f_i(x_{i,t}) = a\log(bx_{i,t}+1)$.

response behavior in Figure 5 assuming the above log-form utility function. When $\theta_t \leq \pi_{i,t}$, i.e., the VM price is lower than unit delay cost, then the tenant can actually admit as many VMs as possible to gain more profit without delaying any VMs. Since the total number of VMs at the beginning of time-slot t is $X_{i,t}$, the tenant can only admit up to $X_{i,t}$ VMs. As the VM price increases, if $\pi_{i,t} < \theta_t \leq \tilde{\theta}_t$, the tenant can still make more profit by increasing number of VMs admitted as its revenue per VM outweighs the net loss of admitting a VM ($\theta_t - \pi_{i,t}$), so the number of VMs admitted is still $X_{i,t}$. $\tilde{\theta}_t$ is the break-even VM price where the tenant's revenue equals to the cost of admitting all VMs. When $\theta_t > \tilde{\theta}_t$, the net cost of admitting a VM increases, thus the tenant decides to reduce the number of VMs admitted.

Our next claim shows the optimal VM pricing decision of the cloud if it has perfect knowledge of the tenant's demand and DR.

Claim 2. *If the cloud has perfect knowledge of the tenants' optimal control decisions, $g(x) = \delta x$, $f_i(x) = f(x) = a\log(bx+1)$, $\pi_{i,t} = \pi_t$, and $X_{i,t} = X_t, \forall i \in [1, N]$, then the cloud's optimal VM price is*

$$\theta_t^* = \begin{cases} \tilde{\theta}_t, & \text{if } \frac{1}{c}\alpha_t\delta \leq \pi_t \\ \max\{\tilde{\theta}_t, \frac{1}{c}\alpha_t\delta\}, & \text{if } \frac{1}{c}\alpha_t\delta > \pi_t, \max\{\tilde{\theta}_t, \frac{1}{c}\alpha_t\delta\} > \sqrt{ab(\frac{1}{c}\alpha_t\delta - \pi_t)} + \pi_t \\ \sqrt{ab(\frac{1}{c}\alpha_t\delta - \pi_t)} + \pi_t, & \text{otherwise} \end{cases}$$

where $\tilde{\theta}_t = \pi_t + f'(X_t) = \pi_t + \frac{ab}{bX_t+1}$.

Here $\frac{1}{c}\alpha_t\delta$ is the cloud's cost of serving one VM. We can make several useful observations from Claim 2. (i) If the cloud's cost per VM is low enough ($\frac{1}{c}\alpha_t\delta \leq \pi_t$), it can always have more profit by serving one more VM, so the best strategy is to set the highest VM price that does not result in tenant's back-off, i.e., tenants admit all of their demand. (ii) π_t reflects the tenant's intolerance to delay; the higher π_t is, the more VMs the tenant has to run, allowing the cloud to increase VM price without losing (too many) VMs. (iii) A tenant with higher a and b will have higher marginal revenue than another tenant with lower a and b. Hence, the cloud can set a higher price for the former tenant. (iv) Since $f'(X_t)$ is decreasing in X_t, $\tilde{\theta}_t$ is also decreasing in X_t, which implies that the cloud might have to reduce its VM price to encourage tenants to run VMs in order to have more profit when $\frac{1}{c}\alpha_t\delta \leq \pi_t$. A **key insight** from Claim 2 is that setting constant VM price or VM price proportional to energy price naively might not be a good choice for the cloud's profit maximization. *The cloud must be careful about exploring the tradeoffs between cloud's energy cost, tenant's demand and DR to optimize its profit.*

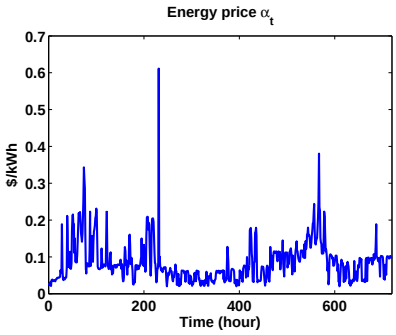


Figure 6: Hourly energy price from an electric utility in [14].

4. EVALUATION

We carry out our evaluation for three "scenarios" in increasing order of complexity. For our simplest scenario I, we only work with a constant energy price and a constant tenant demand partly as a sanity check (Section 4.2). In scenario II, we use real-world energy prices and tenant's demand from an IBM production data center to show the impact of different predictors on cloud's profitability assuming constant delay penalty, i.e., time-invariant tenant DR (Section 4.3). Finally, in scenario III, we also vary tenant's DR over time (Section 4.4).

4.1 Experiment setup

In all the following experiments, we choose to work with "homogeneous" tenants (i.e., having identical DR behavior) from the IBM tenant trace and leave experiments with diverse tenants to future work. Even with this simplification, we still gather interesting insights upon which our future work can build.

Inputs: For scenario I, we work with a constant energy price $\alpha_t = 0.1\$/kWh, \forall t$ and constant tenant VM demand (400 VMs arrive at the beginning of the simulation and stay in the system forever). In scenarios II and III, we use the hourly energy prices in January, 2014 from an electric utility in [14] as shown in Figure 6. Our tenant VM demand trace comes from a production data center operated by IBM and is shown in Figure 2(c) wherein most of the VMs are long-lived and stay in the system during the entire simulation[4].

Parameters: Table 2 presents various parameters. We assume that the cloud can convert tenants' aggregate VM demand into power consumption as $g(x_t) = \delta x_t$, with the complementary assumption that the cloud consolidates ten VMs on each physical server which consumes 500 Watts on average. Then we can compute the power consumption of a single VM as $\delta = 500/(10*1000)*1.2 = 0.06kW/\text{VM}$ assuming the power usage effectiveness PUE = 1.2. We assume that the tenant's utility function has *log*-form: $f_i(x_{i,t}) = a\log(bx_{i,t}+1)$. We explored many combinations of these parameters during our evaluation. Our final choice of the parameters values is based on the following **guiding rule**: all parameters should have non-negligible impact on cloud's

[4]We omit the results from Google trace since the insights are similar with those from the IBM tenant trace.

and tenants' profitability, and all terms in all utilities are non-negligible. We conducted extensive experiments and present here interesting and representative results.

Sym.	Value	Definition
δ	0.06kW/VM	Power consump. of a single VM
c	17%	Frac. of cloud's costs as energy costs
a	10	Param. of tenant's utility function
b	5\$/VM	Param. of tenant's utility function
π	0.05\$/VM	Default delay cost in scenarios I & II

Table 2: Values chosen for our models.

Cloud's predictors: We study the impact of four different predictors employing different degrees of freedom (DoF). Intuitively, predictors with higher DoF should yield higher profits for the cloud since they might provide better prediction of the tenant's aggregate demand for the cloud; however, with higher uncertainty in utility pricing, tenant's demand and DR (particularly when tenant's DR is also time-varying as in Scenario III), it is possible that not all past data are equally valuable for the cloud to tune the parameters of the predictors. Therefore, we choose a "forgetting factor", denoted as λ, as a tuning parameter for the predictors which reflects how quickly the cloud forgets past sample observations of (demand, price) pairs[5]. Larger λ implies that the cloud forgets past data faster. We only show the results under different forgetting factors ($\lambda = 0, 0.1, 0.2$) for the piecewise linear predictor: "PW-M-λ" refers to a piecewise predictor with M line segments and forgetting factor λ.

Tenant's predictor: Generally speaking, tenants might want to predict both their respective VM demand and future VM prices. However, as discussed in Section 3, since we assume that tenant's revenue is only a function of the number of VMs it runs in the current time-slot, the tenant's control problem reduces to optimizing net utility within a single time-slot and thus prediction of future demand is not needed.

Baseline: We choose the case that the cloud has perfect knowledge of both tenant's demand and DR as our baseline to compare with, denoted as "Baseline". As discussed in Section 2, this baseline is sub-optimal w.r.t. long-term profits since it only optimizes the short-term objective. It is possible that some predictors might yield better long-term profits than this baseline for the cloud, however, only by chance, which is verified in our experiment results.

4.2 Constant energy price and tenant's demand

In this section, we assume a constant energy price ($\alpha_t = 0.1\$/kWh$) and constant tenant demand (400 VMs arrive at the beginning of the first time-slot and stay in the system within the entire simulation).

Performance expectations. We expect that the optimal VM price be a constant; all the past data are equally important for the predictors since all the inputs (including tenant's DR) are constant. We expect that the cloud's predictors for tenants' DR with smaller forgetting factors and higher DoF to offer better profits for the cloud.

[5]A forgetting factor can be used to minimize weighted least square error in sample data, e.g., $\min \sum_{t=0}^{n} (1-\lambda)^{n-t}(x_t - \hat{x}_t)^2$, when out-dated data are less useful.

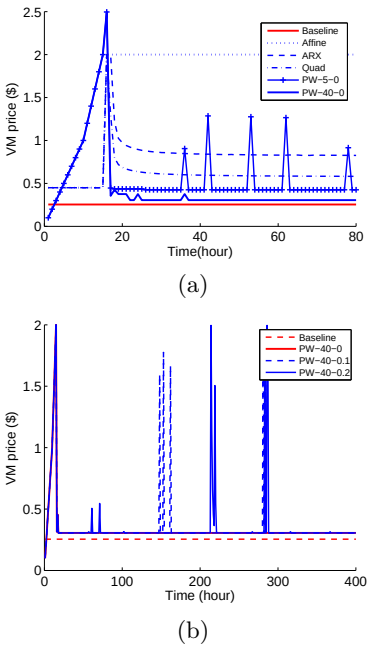


Figure 7: Scenario I: (a) VM prices under different predictors with $\lambda = 0$. (b) VM prices under piecewise linear predictors with different forgetting factors. PW-40-0 overlaps with others but without spikes.

Figure 7(a) shows the VM prices under different predictors with $\lambda = 0$. We observe that the VM price under simple predictors (ARX, Affine and Quad) with smaller DoF converges to a sub-optimal price (Baseline is optimal w.r.t. the above utilities) whereas error in estimate of aggregate tenant DR is minimized when the tenant demand (and DR) is time-invariant and the cloud knows this to be the case and simply tabulates past (price,demand) =(θ_t,x_t) observations and interpolates between them. As an example, the PW-40-0 performs much better than PW-5-0. We also observe that the steady-state VM price converges towards the optimal price very quickly as we increase DoF, i.e., the number of line segments (though results not shown here). However, when the aggregate tenant DR is time-varying (possibly due to tenant churn), system identification (to tune the parameters of the predictors) with aged-out data (i.e., forgetting factor larger than 0) is required.

We show the VM prices under the same piecewise linear predictor with different forgetting factors in Figure 7(b). We find that when $\lambda = 0$, the VM price converges very quickly to a sub-optimal price and stays stable. As λ increases from 0 to 0.2, the VM price exhibits more fluctuations. This is because all inputs (including tenant's DR) are time-invariant, and that all past data are important for the cloud's estimation of tenant's DR. Thus the performance of the predictor gets better as λ increases, but fluctuations also increase.

Key insights: (i) DoF affects the optimality of the predictors and higher DoF offers better performance. (ii) Convergence to sub-optimal price is faster for smaller forgetting factor. When inputs are constant and tenant's DR is time-invariant, larger forgetting factor results in oscillation/unstable behavior whereas smaller forgetting factor

yields faster convergence and maintains stability.

4.3 Real-world energy prices and tenant demand

In this section we look at the scenario with real-world energy price and tenant's demand as introduced in Section 4.1.

Performance expectations. We expect to see that the cloud has better profits under predictors with higher DoF. However, due to the poor predictability of utility prices and tenants' demand, it is possible that not all past data are equally useful for the cloud's estimation of tenant and higher forgetting factor might offer better profit.

Figure 8 shows the VM prices, cloud's cumulative profit and tenant's cumulative profit over a month with different predictors and $\lambda = 0$. First, we observe that the "Baseline" assuming perfect knowledge of tenant's demand and DR offers dynamic VM prices which have similar fluctuation with the energy prices α_t. However, when the energy price is low, the VM price seems stable and does not depend on energy price. This is consistent with Claim 2 in Section 3 that only when energy price is high enough ($\frac{1}{c}\alpha_t\delta > \pi_t$), the optimal VM price is either proportional to energy price, or depends on the tradeoffs between energy price, tenant's utility and delay cost. Second, we find that VM prices generated under predictors with low DoF (such as ARX, Affine and Quad) converge to different constant VM prices and there is great gap between the cloud's profits with those predictors and the baseline profit. This justifies our motivation for DR-aware dynamic pricing: Simply setting **constant VM prices** cannot guarantee cloud's profitability. On the other hand, predictors with higher DoF (such as PW-40-0) offer much better profit for the cloud (higher than Affine by 30%), with VM prices closer to those of Baseline. Third, tenant's profit could be negative (as shown in Figure 8(b)) due to high VM price. As an extreme case, the VM price under Quad is so high that the tenant has to defer most of its VMs which results in high delay cost and even negative profit. In such cases, the tenant might have to switch to a different cloud provider (an action space beyond the scope of this paper).

Next, we evaluate the impact of different forgetting factors. As we expected, in Figure 9(a), PW-40-0.1 and PW-40-0.2 offer similar cumulative profit for the cloud, which is much higher than that of PW-40-0. This is due to the poor predictability in both the utility price and tenant's demand. In such cases, out-dated past samples of (price, demand) might not be so useful for the cloud to estimate tenant's DR, consistent with the choice of high forgetting factor. However, as λ keeps increasing, the cloud's profit starts to decrease. This is due to the fact that predictors with higher λ might be too responsive in estimating tenant's DR, and oscillating behavior (in the VM prices) might occur which may in turn hurt the cloud's profitability.

In addition, we explore the impact of delay cost π on both the cloud's and tenant's profits by varying $\pi = 0.025, 0.05$ while keeping all other parameters and inputs the same. In Figure 9(b), the cloud achieves higher profit when the tenant's delay cost is larger. This is consistent with our intuition: larger delay cost implies that the tenant is willing to pay higher price for the same amount of VMs to guarantee its workload performance, and thus the cloud could make more profit by charging higher price without losing much VM demand, whereas a tenant with less delay cost has more flexibility in its demand and thus the cloud's profit becomes less. Intuitively, the opposite should happen at the tenant-side: as delay cost increases and the cloud makes more profit, the tenant's profit should decrease. However, as shown in Figure 9(c), the tenant's profits under different delay costs are quite close. This is due to the fact that the cloud's price is sensitive to π (as in Claim 2): higher delay cost causes higher VM price, and the increase of tenant's delay cost can be canceled by the increase of VM price to some extent (recall that $x^*_{i,t} = \frac{a}{\theta_t - \pi_{i,t}} - 1/b$ when $\theta_t \leq \tilde{\theta}_t$).

Key insights: (i) Similar to scenario I, predictors with higher DoF offer better profit for the cloud. (ii) However, forgetting factor has to be chosen carefully (instead of using $\lambda = 0$ in scenario I) due to the poor predictability of inputs. (iii) Setting a constant VM price does poorly for the cloud's profit compared with the proposed dynamic pricing.

Remark. In scenario II, the Baseline always offers more profit for the cloud than other predictors. However, recall that our myopic control only maximizes short-term profit as discussed in Section 2, which is not the way we assess the performance (i.e., long-term profit of the cloud) of different predictors. In fact, we can even create scenarios wherein the Baseline solution is not optimal w.r.t. the cloud's long-term profit. We show such results in Section 4.4.

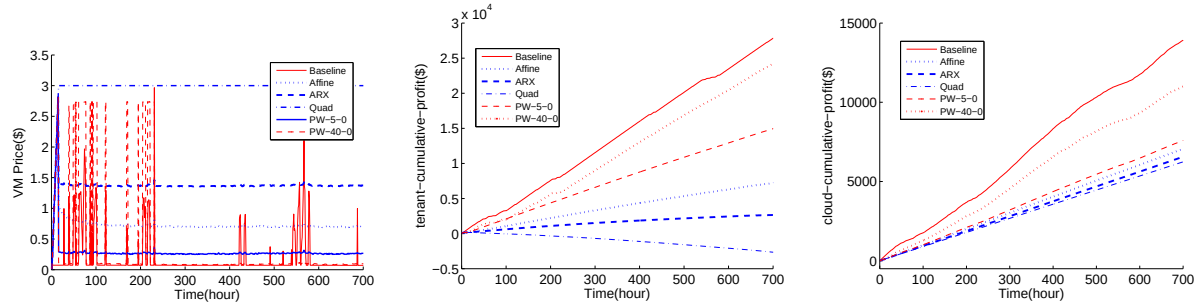
4.4 Real-world data and time-varying DR

In this section we look at the scenario with assumptions in scenario II. We introduce more dynamic behavior at tenant side by imagining time-varying delay cost. We vary the tenant's delay cost π_t between $0.005\$/VM$ and $0.1\$/VM$ every 48 time-slots[6].

Performance expectations. Time-varying delay costs add to the complications of prediction at cloud side. Here based on insights from section 4.3, we expect to see better performance for predictors with higher DoF. Different λ values are used to see the effect of forgetting past data. In the next section, we study the non-trivial trade-off associated with the choice of forgetting factor λ.

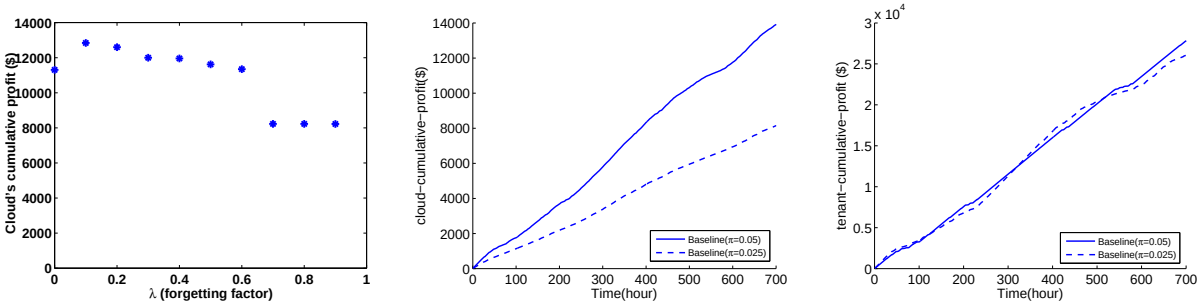
We observe in our experiments that predictors with lower DoF convergence to a constant price which does not perform well in terms of cloud profit. This is consistent with what is observed in previous section. Therefore, we use the piecewise-linear predictor with 40 segments to predict the highly variable tenants' DR at cloud side. Additionally we study the effect of forgetting the past data to explore this effect in more complex and closer to real-world scenarios. The

[6]In the real world, it is possible that the tenants might have highly variable delay costs or utility parameters. For such cases, we can have a data collection system before applying the pricing framework for system identification. For example, we can filter the past samples and only look at the (price, demand) pairs at, e.g., the same time-of-day, energy price and delay penalty regime, to tune the parameters of the cloud's predictors. We leave this line of study to future work.



(a) VM prices (b) Tenant's cumulative profit. (c) Cloud's cumulative profit.

Figure 8: Scenario II with different predictors and $\lambda = 0$.



(a) (b) (c)

Figure 9: Scenario II: (a) Cloud's cumulative profit under PW-40-λ, $\lambda \in [0, 0.9]$. (b) Cloud's cumulative profit under Baseline with different delay costs $\pi = 0.025, 0.05$. (c) Tenant's cumulative profit under Baseline with $\pi = 0.025, 0.05$.

piecewise predictor which was observed to have the best performance in previous sections is chosen. In section 4.3 the myopic objective gives the best performance for the cloud but here we see that under highly variable situations (which is close to real-world cases) satisfying myopic objective does not guarantee the end goal which is cloud cumulative profit. In Figure 10, we show that the cloud's profit using PW-40-0 is higher than Baseline by around 40% - this is possible because the myopic objective of the cloud and tenant does not guarantee optimal long-term profit (as a longer-term objective would under standard Markovian assumptions).

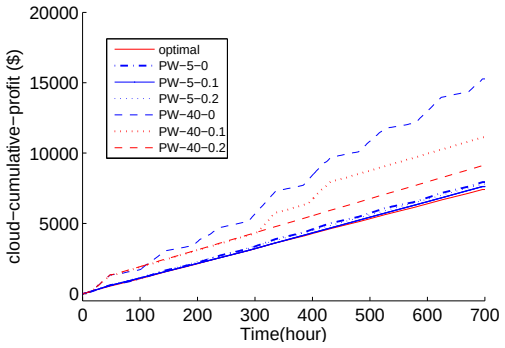


Figure 10: Scenario III: PW with different forgetting factors.

Key insights: (i) Predictors with higher DoF perform better even in highly variable environments. (ii) Optimal solutions of myopic control does not guarantee long term optimal profit for the cloud.

5. RELATED WORK

Pricing design in the cloud. Pricing in the cloud has emerged as an active area of research, and various techniques including dynamic pricing [17, 30, 22], auction-based pricing [26, 20], Nash bargaining [7] with either single or federation among cloud providers [8, 3], have been proposed. However, a common assumption of existing work is the tenants' DRs/demands can be inferred by the cloud, generally for theoretic tractability reasons, which is not suitable in our work due to poor predictability of the real-world data we use and the complexity of tenant's DR. We do find one exception: in [23], the cloud designs pricing for bandwidth reservation without knowing tenants' DR and decentralized algorithms are proposed with optimality guarantees. However, such nice properties largely depend on the way that the problem is constructed, e.g., [23] maximizes the social welfare of the cloud and the tenants which is not a reasonable goal in our environment due to the assumed selfish nature of these entities.

Game with incomplete information. Although not well-explored in cloud computing yet, in other areas (e.g., power systems) general incomplete-information game-theoretic frameworks include Bayesian games and hypergames, e.g., [25], have been explored. In both VCG and PSP auctions (e.g., [4] recently), issues of truthfulness in the disclosed bids are considered, i.e., reflecting actual demand response (by marginal valuation). More prosaic approaches simply interpolate and extrapolate from presumed honest bids (by (amount, price)) to obtain a complete estimate of other players' demand response. These frameworks are applicable to iterated (sequential) adversarial (non-cooperative) games with or without leaders. Generally, estimates are greatly

simplified under the assumption that player strategies are time (play-action iteration) invariant. In future work, we will consider introducing such more complex/advanced techniques to our cloud pricing design.

DR in cloud computing. A large body of work now exists on DR for data centers that is complementary to our work. A comprehensive survey is offered in [29]. DR by individual tenants, as imagined in our work, is relatively less well-explored.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a leader-follower game-based cloud pricing framework for cloud's profit maximization. In face of poor predictability of tenants' demand and DRs, we employed myopic control with short-term predictive models. Our empirical evaluation with real-world data showed useful insights on cloud's profitability affected by its predictive models. In our future work, we will explore enhancements to our models identified throughout our paper and carry out more comprehensive evaluation using benchmarks and realistic prototypes.

7. REFERENCES

[1] Amazon EC2 Pricing, 2014. http://aws.amazon.com/ec2/pricing/.

[2] L. André Barroso and U. Hölzle. *The Datacenter as a computer: an introduction to the design of warehouse-scale machines*. Morgan & Claypool, 2009.

[3] J. Anselmi, D. Ardagna, J. C. S. Liu, A. Wierman, Y. Xu, and Z. Yang. The economics of the cloud: price competition and congestion. *SIGMETRICS Perform. Eval. Rev.*, 41(4), April 2014.

[4] S. Bhattacharya, K. Kar, J. H. Chow, and A. Gupta. Progressive second price auctions with elastic supply for pev charging in the smart grid. In *Proc. of NeTGCoop*, 2014.

[5] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proc. of NSDI*, 2008.

[6] Y. Chen, A. Ganapathi, R. Griffith, and R. H. Katz. The case for evaluating mapreduce performance using workload suites. In *Proc. of IEEE MASCOTS*, 2011.

[7] Y. Feng, B. Li, and B. Li. Bargaining towards maximized resource utilization in video streaming datacenters. In *Proc. of INFOCOM*, 2012.

[8] Y. Feng, B. Li, and B. Li. Price competition in an oligopoly market with multiple iaas cloud providers. *IEEE Trans. Computers*, 63(1), 2014.

[9] M. Ghorbani, Y. Wang, Y. Xue, M. Pedram, and P. Bogdan. Prediction and control of bursty cloud workloads: a fractal framework. In *Proc. of CODES*, 2014.

[10] Google cluster data, 2011. https://code.google.com/p/googleclusterdata/.

[11] D. Hamilton. Most US companies plan to increase public cloud spending 15 percent or more in 2015. http://www.thewhir.com/web-hosting-news/us-companies-plan-increase-public-cloud-spending-15-percent-2015.

[12] J. Hamilton. Internet-scale service infrastructure efficiency. *SIGARCH Comput. Archit. News*, 37(3):232–232, June 2009.

[13] Y. He, S. Elnikety, J. R. Larus, and C. Yan. Zeta: scheduling interactive services with partial execution. In *Proc. of SoCC*, 2012.

[14] Ontario electric utility, 2014. http://www.ieso.ca/Pages/Power-Data/default.aspx#price.

[15] D. Juan, L. Li, H. Peng, D. Marculescu, and C. Faloutsos. Beyond poisson: modeling inter-arrival time of requests in a datacenter. In *Advances in knowledge discovery and data mining*. Springer International Publishing, 2014.

[16] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The nature of data center traffic: measurements & analysis. In *Proc. of ACM SIGCOMM*, 2009.

[17] V. Kantere, D. Dash, G. Francois, S. Kyriakopoulou, and A. Ailamaki. Optimal service pricing for a cloud cache. *IEEE Trans. Knowledge and Data Engineering*, 2011.

[18] R. T. Kaushik, P. Sarkar, and A. Gharaibeh. Greening the compute cloud's pricing plans. HotPower, 2013.

[19] W. Kwon and S. Han. *Receding horizon control*. Invited chapter for Green High-Performance Computing. SpringerLink, 2005.

[20] H. Li, C. Wu, Z. Li, and F. Lau. Profit-maximizing virtual machine trading in a federation of selfish clouds. In *Proc. of IEEE INFOCOM*, 2013.

[21] D. Link. Netflix and stolen time. http://blog.sciencelogic.com/netflix-steals-time-in-the-cloud-and-from-users/03/2011.

[22] Z. Liu, I. Liu, S. Low, and A. Wierman. Pricing data center demand response. In *Proc. of ACM SIGMETRICS*, 2014.

[23] D. Niu, C. Feng, and B. Li. Pricing cloud bandwidth reservations under demand uncertainty. In *Proc. of ACM SIGMETRICS*, 2012.

[24] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch. Heterogeneity and dynamicity of clouds at scale: google trace analysis. In *Proc. of ACM SoCC*, 2012.

[25] Y. Sasaki and K. Kijima. Hypergames and bayesian games: A theoretical comparison of the models of games with incomplete information. *J Syst Sci Complex*, 25:720–735, 2012.

[26] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C.M. Lau. An online auction framework for dynamic resource provisioning in cloud computing. In *Proc. of ACM SIGMETRICS*, 2014.

[27] R. Vartabedian. US electricity prices may be going up for good. http://www.latimes.com/nation/la-na-power-prices-20140426-story.html.

[28] C. Wang and et. al. Recouping energy costs from cloud tenants: Tenant demand response aware pricing design. Technical report, CSE TR-15-001, Penn State University. http://www.cse.psu.edu/~bhuvan/CSE-TR-15-001.pdf.

[29] A. Wierman, Z. Liu, I. Liu, and H. Mohsenian-Rad. Opportunities and challenges for data center demand response. In *Proc. of IEEE IGCC*, 2014.

[30] H. Xu and B. Li. Maximizing revenue with dynamic cloud pricing: the infinite horizon case. In *Proc. of IEEE ICC*, 2012.

[31] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F.C.M. Lau. Dynamic pricing and profit maximization for the cloud with geo-distributed data centers. In *Proc. of IEEE INFOCOM*, 2014.