Dynamic Budget-Constrained Pricing in the Cloud

Authors [names removed for double-blind review]

Affiliations [removed for double-blind review]

Abstract. We introduce a new analytical model of user-based dynamic pricing in which decisions occur in real time and are strongly influenced by the budget constraints of users. This model captures the fundamental operation of many electronic markets that are used for allocating resources. In particular, we focus on those used in data centers and cloud computing where pricing is often an internal mechanism used to efficiently allocate virtual machines. We study the allocative properties and dynamic stability of this pricing model under a standard framework of cloud computing systems which leads to highly degenerate systems of prices. We show that as the size of the system grows the user-based budget-constrained dynamic pricing mechanism converges to the standard Walrasian prices. To show this, we consider the "quasi-static" approximation for the model that provides tractability for theoretical analysis. However, for finite systems, the prices can be non-degenerate and the allocations unfair, with large groups of users receiving allocations significantly below their fair share. In addition, we show that improper choice of price update parameters can lead to significant instabilities in prices, which could be problematic in real cloud computing systems, by inducing system instabilities and allowing manipulations by users. We construct scaling rules for parameters that reduce these instabilities.

1 Introduction

Price-based mechanisms provide simple, powerful, and robust tools to allocate resources in complex systems. They are easy to design, as they adaptively set prices for each resource and then allow users to purchase their optimal bundle of resources at those prices. Furthermore, unlike traditional algorithmic methods, they adapt easily to changes or additions in the architecture of the underlying system. For these reasons they are widely used for internal pricing to optimize resource allocation in cloud computing and data centers.

While the static/equilibrium theory of price mechanisms is well established, their dynamics are not as well understood. However, in modern electronic markets and computer systems the real-time behavior is crucial. The most well understood dynamics of price mechanisms are studies of the Tatonnement [17], a fictitious price adjustment mechanism where users reveal their true preferences to a sequence of hypothetical prices. While this and other previously studied dynamic models of pricing (e.g., [7, 4]) can be informative, they do not capture the key issues that arise in many computational settings. They both overlook important dynamical details as well as ignoring budget constraints—an important aspect of real systems using user-based markets.[1]

In this paper we construct a dynamic model of real time pricing that is driven by users' budget constraints. We show that in the limit of a large number of users (relative to the number of different types of resources) budget-constrained pricing (BCP) leads to the standard (static) price equilibrium. However, with a finite number of users there are discrepancies and instabilities which we study both analytically and via simulations.

For concreteness, we focus on a specific resource allocation problem arising in data centers and cloud computing: allocating virtual machines (VMs) to users, where the goal is to simultaneously maintain high efficiency and max-min fairness [9]. Unlike some well-known systems like Amazon's Elastic Cloud Compute Center (EC3) where VMs are rented out to maximize profit, and allocative fairness is secondary, these clouds are typically either used by a single large organization, where max-min fairness is based on internal divisions in the company [2], or shared by many organizations over the long term and max-min fairness ensures that each receives its correct share of the resources [11, 9].

In such systems (e.g., [12]) there are typically tens to hundreds of users and thousands to millions of VMs. The key constraint is that the VM be compatible with the user's request. For example, many users have tasks that can only run under a specific operating system (Linux/Windows), a VM with specific hardware, such as a GPU or other properties, such as a public IP address.

These user-based markets use scrip [14, 15, 3, 7] (money that has no value outside the system) which is replenished at a regular rate. Given this dynamic supply of scrip, budgeting is of primary importance and budget constraints can dominate user behavior, i.e., if a user spends all its money it cannot run any more jobs until a replenishment arrives.

We study the behavior of these user-based budget-constrained markets and show that they are surprisingly effective—they can attain both high efficiency and max-min fairness. This is true in the large market limit, but can also be attained for finite markets subject to some key design principles that we uncover.

In particular, our analysis shows that these user-based budget-constrained markets provide a solution to a problem that arises in the analysis of these systems using a classical market. This problem arises because in equilibrium the classical Walrasian prices are highly degenerate for this economy. In particular, the equilibrium prices in such an economy are all the same which can lead to problems in allocations, i.e., when prices of different VMs are the same but the optimal allocation requires a specific allocation, it is unclear how users would implement this allocation when they are unaware of its details.

We show that this problem is largely resolved by the budget-constrained behavior which leads to slight differences in prices providing the incentives to move

[1] The importance of budgetary constraints in auction design is also an important topic of current study [6, 10].

away from over-demanded VMs. These price differences can be either transitory in the case of tie breaking or permanent in the case of more popular machines.

Our model is also of interest to the study of pricing and markets in economic theory as it provides a new alternative to the Tatonnement. In particular, our "quasi-static" approximation provides a tractable model of price adjustments that may be more widely applicable.

2 Model

Our model consists of two parts: (1) a standard model of internal cloud computing centers (CCCs) where VMs are allocated to users, coupled with (2) a user-based budget-constrained dynamic pricing mechanism, which governs the prices of VMs, and consequently also affects the allocation of VMs to users. The main contribution of our paper is studying the effects of this user-based budget-constrained dynamic pricing mechanism.

Let us first introduce the standard model of CCCs that we consider. The set of VMs, M, is divided into classes according to the partition C, and let $m = |M|$. The set of users is denoted by N, with $n = |N|$. Each user i has a set of allowable VM classes, $C_i \subseteq C$, on which she can run her tasks. In order to avoid trivialities, assume that every class is demanded by at least one user.

As shown in Fig. 1, these restrictions create a bipartite graph between users and VMs: circles represent groups of users with the same sets of allowable VM classes, squares represent the classes of VMs, and edges connect the groups of users to their allowable types. For example, Fig. 1a shows a system with $k+1$ classes of VMs and $k+1$ groups of users, where users of type 0 only like class 0 VMs, while for $1 \leq j \leq k$ users of type j like VMs from classes 0 and j. We refer to this as the "finicky" example, as users in group 0 only want the overdemanded VMs. Fig. 1b shows a model with more complex preferences which we will consider later.

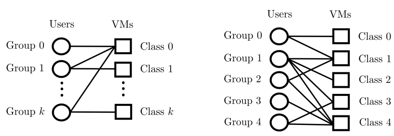

Fig. 1: **Bipartite graph that illustrates preference structures and can be used to compute feasibility of max-min allocations.**

Let X represent an allocation of VMs where $X_i(t)$ is the set of VMs allocated to user i at time t, and let $x_i(t) = |X_i(t)|$. For analytic tractability[2], we assume

[2] We admit that this assumption is at odds with standard run time distributions of tasks in CCCs, which tend to be heavy tailed [1], but believe that by greatly

that all tasks take the same amount of time on average and are distributed i.i.d. according to an exponential distribution with rate α. We assume that users receive equal utility for each completed job, and are indifferent between different VMs within their allowable types. Consequently, $x_i(t)$ is the instantaneous utility of user i at time t, since we assume that users are only allocated their allowable classes and we are focusing on the average value over time of this. In particular, we will be interested in the equal utility allocation, $x_i(t) = m/n$, and more generally the max-min fair allocations [11, 9]. Note that one can easily check the feasibility of the max-min fair allocations (or any other specified allocation) by solving a maximum flow on the augmented bipartite graph in Fig. 1.

In order to dynamically allocate the VMs we consider a simple dynamic pricing mechanism that captures the essence of many pricing mechanisms used in CCCs. Unlike traditional (non-price-based) allocation mechanisms for CCCs (see [9] for an examplar and the references therein), user-based pricing mechanisms are simple decentralized mechanisms that can easily adapt to changes in the system, as the optimization is facilitated directly by the users. However, in addition to the added responsibility this imposes on users, it may also lead to instabilities, as we will see later.

In our model we assume that each VM class c has a current price $p_c(t)$ at time t. Users receive allotments, s units, of scrip at small time intervals (the lengths of which we assume for tractability are i.i.d. exponentially distributed with rate γ), and their current budget at time t is given by $b_i(t)$.

When a VM completes a task it becomes available and users can request to run their next task on that VM. In order to focus on budgetary issues rather than strategic purchasing decisions, we make the simplifying assumption that a user will always request a VM if the class c of this VM is allowable for them, i.e., $c \in C_i$, and if they have sufficient budget, i.e., $b_i(t) \geq p_c(t)$. In particular, this assumes that users are not constrained by capacities—they want as many machines as possible; they are constrained only by their budgets. Let $U(t)$ be the set of users requesting a VM from class c that became available at time t, and let $u(t) = |U(t)|$. If $u(t) > 0$ then the VM is allocated uniformly at random to one of the users in $U(t)$, and the chosen user's budget is decreased by $p_c(t)$ (i.e., they pay for the machine immediately); otherwise the VM becomes inactive for an exponentially distributed time with rate β. In both cases the price is updated as described below. When an inactive VM becomes available again it is readvertised to the users, some of which may now have enough budget to purchase the VM. Note that the use of a finite readvertising rate simplifies the pricing update rates allowing one to update a finite number of times at specified (stochastic) intervals. It also guarantees that at most one VM becomes available at a time.

The price is updated according to a simple multiplicative rule:

$$p_c(t) \to \exp(\epsilon(u(t) - 1))p_c(t), \tag{1}$$

facilitating the analysis and providing insights into problems that would not be tractable otherwise, it provides a useful approximation.

where $\epsilon > 0$. This specific rule is chosen for its tractability and its value in constructing polynomial time algorithms [4], but does not affect our main results as long as the price decreases for $u(t) = 0$ and increases for $u(t) > 1$. For example, one could use an additive rule, as is common in the economics literature [16]: $p_c(t) \to p_c(t) + \epsilon(u(t) - 1)$. For such rules, if a class of VMs is consistently not demanded then its price will fall, while if a class of VMs is overdemanded, then its price will rise.

3 Theoretical Results

The full theoretical analysis of the system described in Section 2 appears quite complex. Nonetheless, we are able to analyze various simplifications of the model, which provide us with useful intuition on what behavior to expect of the system; these intuitions are then confirmed via simulations in Section 4.

3.1 A single class

To understand the basic behavior of our model we first provide an analysis of a CCC with a single class of VMs. In order to understand the equilibrium prices we consider the limit of slow price adjustments (i.e., $\epsilon \approx 0$). To see these analytically, we fix the price (set ϵ to 0) and find conditions under which the effect of price adjustments, at that price, would average out, leading to no macroscopic drift.

Recall that there are n users, m VMs, with task completion rate α, idle VM readvertising rate β, and scrip replenishment rate γ. In order to reduce the size of the state space we assume that the prices are the same as the budget replenishment sizes, $p = s = 1$, since then the vector of budgets $b(t)$ is integral, allowing us to relate the budget dynamics to a random walk on a lattice. Rather than vary p directly, we note that the key ratio is $p\alpha/(\gamma s)$ and thus, due to rescaling, we only need to consider finding the value of γ for which the price is in equilibrium.

Next we note that the number of nonzero entries of $b(t)$ is exactly $u(t)$ and let $R(t)$ be the number of idle VMs at time t (i.e., the size of the reserve). Then $(b(t), R(t))$ follows a continuous-time random walk in $\mathbb{N}^n \times \{0, 1, \ldots, m\}$ with the following rates. For every i, at rate γ, $(b(t), R(t)) \to (b(t) + e_i, R(t))$, where e_i is the unit vector in direction i; if $b_i(t) > 0$, then $(b(t), R(t)) \to (b(t) - e_i, R(t))$ at rate $\alpha(m - R(t))/u(t)$ and $(b(t), R(t)) \to (b(t) - e_i, R(t) - 1)$ at rate $\beta R(t)/u(t)$; and finally if $b(t) = 0$, then $(0, R(t)) \to (0, R(t) + 1)$ at rate $\alpha(m - R(t))$.

In fact, denoting $|b(t)| = \sum_{i=1}^{n} b_i(t)$, it follows that the pair $(|b(t)|, R(t))$ is also a Markov chain, with state space $\mathbb{N} \times \{0, 1, \ldots, m\}$. For $m = 1$, the stationary distribution π of this chain (assuming $\gamma n < \alpha, \beta$, so that it is positive recurrent) can be computed explicitly. We have for $i \geq 1$ (calculations omitted):

$$\begin{aligned}
\pi(i,0) = \pi(0,0) \Bigg\{ & \frac{\alpha^2 \beta (\beta + \gamma n)}{(\gamma n)^2 (-\alpha + \beta + \gamma n)(\alpha - \gamma n)} \times \\
& \times \left(\frac{\gamma n}{\alpha}\right)^{i+1} - \frac{(\gamma n)^2 + \alpha\beta}{\gamma n (\alpha - \gamma n)} \left(\frac{\gamma n}{\alpha}\right)^i \\
& - \frac{\alpha (\beta + \gamma n)^2}{(\gamma n)^2 (-\alpha + \beta + \gamma n)} \left(\frac{\gamma n}{\beta + \gamma n}\right)^{i+1} \Bigg\}, \\
\pi(i,1) = \pi(0,1) & \left(\frac{\gamma n}{\gamma n + \beta}\right)^i,
\end{aligned}$$

where $\pi(0,1) = \frac{\alpha}{\gamma n}\pi(0,0)$ and

$$\pi(0,0) = \frac{\gamma n}{\alpha} \frac{\beta (\alpha - \gamma n)^2 (-\alpha + \beta + \gamma n)}{\alpha (\beta - \gamma n)(\alpha - \gamma n)^2 - (\gamma n)^2 \beta (-\alpha + \beta + \gamma n) + \alpha^2 \beta^2}.$$

From this it follows that the fraction of time the VM is unused, $\sum_{i=0}^{\infty} \pi(i,1)$, is equal to

$$\frac{(\gamma n + \beta)(\alpha - \gamma n)^2 (-\alpha + \beta + \gamma n)}{\alpha (\beta - \gamma n)(\alpha - \gamma n)^2 - (\gamma n)^2 \beta (-\alpha + \beta + \gamma n) + \alpha^2 \beta^2}.$$

If β is large and $\gamma n / \alpha = 1 - \delta$, then this fraction is approximately $\delta/2$, i.e., the system can be made efficient with proper choice of the parameters.

The condition for an equilibrium price is that if we incorporate price updates with small ϵ, the expected change in price is zero. By taking the logarithm in the price update equation (1), this is equivalent to the condition that $\mathbb{E}(u(t)) = 1$. Though $(|b(t)|, R(t))$ does not contain enough information to compute this directly, in the large n limit each user is extremely unlikely to have a budget larger than 1, so we can approximate $u(t)$ by $|b(t)|$ to approximately compute the equilibrium price as a function of the parameters. After some algebra, the approximate equilibrium condition reduces to

$$\begin{aligned}
\alpha^2 \beta^3 (\beta + \gamma n) + (\beta + \gamma n) \beta (\alpha - \gamma n)^3 (-\alpha + \beta + \gamma n) & \\
- \alpha \beta^2 \left((\gamma n)^2 + \alpha \beta\right) (-\alpha + \beta + \gamma n) & \\
- \alpha (\beta + \gamma n)^2 (\alpha - \gamma n)^3 & \\
= \frac{\beta^2 (\alpha - \gamma n)^3 (-\alpha + \beta + \gamma n)}{\alpha \pi(0,0)} & .
\end{aligned}$$

For larger m this approach becomes unwieldy. However, when $\alpha = \beta$, the dynamics of $b(t)$ becomes autonomous: it performs a continuous-time random walk on $\mathbb{N}$ with jump rate γn upwards and αm downwards. Provided $\gamma n < \alpha m$, this has a geometric stationary distribution with parameter $(\gamma n)/(\alpha m)$.

The intuition to take away from this section is that it is natural to have β larger than α, in order to not have idle machines, i.e., to increase efficiency. The dynamics of the reserve $R(t)$ is not autonomous: in fact, $R(t)$ can only increase when $b(t) = 0$, and it can only decrease when $b(t) \neq 0$. We have $R(t) \to R(t) + 1$ at rate $\alpha(m - R(t))$ when $b(t) = 0$, and $R(t) \to R(t) - 1$ at rate $\beta R(t)$ when $b(t) \neq 0$. We can thus see that large β increases the drift towards zero for $R(t)$, and thus leads to greater efficiency. See also Section 4, which confirms this property for general systems via simulations.

3.2 Quasi-Static Approximations, Asymptotic Fairness and Walrasian Equilibrium

The first key property of price mechanisms is their efficiency; however in the general setting of CCCs efficiency is easy to attain—just allocate any available VM to any user who can use it. So the main reason CCCs use pricing mechanisms is equity (fairness). In our setting the standard (static) Walrasian equilibrium [17] attains the equal share solution if it is feasible, or more generally, it attains the max-min fair solution. We now argue that in the large n limit the equilibrium from user-based budget-constrained dynamic pricing also does so.

Recall that the Walrasian equilibrium [17] is defined as follows: Let m_c be the number of VMs of class c available. User i has $b_i = 1$ units of scrip which she uses to purchase as many VMs as she can afford at the current prices p. Let $X_i(p)$ be the set of VMs that user i purchases at price p, and let $\Xi_i(p)$ be the set of all optimal purchase sets. A price allocation (p, X) is an equilibrium if $X_i(p) \in \Xi_i(p)$, i.e., all users are purchasing an optimal bundle, and $\bigcup_i X_i(p) = M$ with $\bigcap_i X_i(p) = \emptyset$, i.e., all VMs are allocated, with each VM allocated to a single user. One can easily compute in polynomial time the Walrasian equilibrium allocation and prices by solving a max-flow on the augmented bipartite graph in Fig. 1.

Although the utilities in the Walrasian equilibrium allocation are unique [4], note that the equilibrium prices are typically highly degenerate.

Theorem 1. *If VM classes c and c' are both demanded by at least two users and there exists a user who purchases VMs from both class c and c' in a Walrasian equilibrium, then $p_c = p_{c'}$.*

Proof. The fact that c and c' are demanded by at least two users guarantees that the equilibrium prices for these classes are positive: $p_c, p_{c'} > 0$. Then if, say, $p_c > p_{c'} > 0$, then it is suboptimal for user i to request goods from class c; she should simply make all her purchases from the lower priced class c'.

Thus, in equilibrium the prices provide little information. This raises the question of how the users 'know' which classes of VMs to purchase. For example, in the "finicky" example of Fig. 1a, in equilibrium users of group $j > 0$ typically purchase VMs from classes 0 and j. However, since the equilibrium prices are the same for both classes, they have no way of "knowing" how to divide their demands among the two classes of VMs. For example, if there are only 2 classes, i.e., $k = 1$, and there are 100 machines in each class and 10 users in each group then the type 1 users should each purchase exactly 10 VMs in class 1. However, if there were 150 machines in class 0 and 50 in class 1 then the prices would be unchanged (and still equal) yet somehow the type 1 users would be expected to purchase 5 VMs in each class. In more complex situations (e.g., Fig. 1b) this issue is exacerbated, and one would not expect users to purchase the correct bundles relying solely on price information.

This price degeneracy also arises in the user-based budget-constrained dynamic pricing model in the large n limit under suitable parameter choices, which we discuss now. A crucial quantity for the smooth running of user-based budget-constrained dynamic pricing is the rate at which machines appear and are available to be bought by the users. The total rate is $\alpha\,(m - R(t)) + \beta R(t)$, where $R(t)$ is the reserve size at time t, which comes from jobs finishing and idle machines being readvertised. Since this rate is $\Omega\,(m)$, in order to avoid drastic price fluctuations, it is natural to scale the price adjustment constant as $\epsilon = \tilde{\epsilon}/m$. This leads to the price changing a constant amount over a constant amount of time, provided that $\beta R(t) = O(m)$ and $\tilde{\epsilon} = \Theta(1)$. If $\beta R(t) = \omega(m)$, then the prices change (in fact, decrease) rapidly, adjusting to the budget shortage of the users. Thus choosing $\beta = \Theta(m)$ leads to the reserve being of constant size, i.e., this leads to *efficiency*.

The main observation is that the stochastic process describing our dynamic pricing model has *two time-scales*. On the *"fast time-scale"* t, the machine allocations fluctuate, but the prices are essentially unchanged; while on the *"slow time-scale"* $\tau = t/\epsilon$, the prices also change by a constant amount. There has been significant work on multi-time-scale stochastic processes [18]. The main results are that, under suitable conditions, (i) for the fast-time dynamics, the quantity that changes over the slow time-scale can be considered constant, while (ii) for the slow-time dynamics the quantity that changes over the fast time-scale can be "averaged out", i.e., we can assume it is in its stationary distribution.

The state-of-the-art literature for multi-time-scale stochastic processes (see [18] and the references therein) can deal with processes on a countable state space, and also diffusions. However, processes which combine jumps and diffusions have received little attention to date (see [13]), and their multi-time-scale behavior is as of yet not well understood. In our dynamic pricing model prices change often by small amounts, while allocations comprise a jump process, and so it falls into the latter category with no known results on systems of this type. A complete rigorous analysis of dynamic pricing is outside of the scope of the present article, but based on analogies with known simpler systems, we make conjectures about its long-term behavior, supported by heuristic arguments. These are further supported by simulation results in Section 4.

In order to analyze the large system size limit, we consider the following limiting process using replica economies [5]. Given some CCC with m, n, C and C_i's, we define the d-replicant of the economy by taking d copies of every user, with the same preferences, and d copies of every VM, of the same class. This provides a large economy which has essentially the same Walrasian equilibrium, with same prices and allocations for each VM class and user.

For fast time-scales, we can consider the prices as fixed and compute the average demand for each good under this assumption. From this we can estimate the rate and direction of the price changes at the current set of prices, which allows us to construct a vector field of price adjustments to understand the slow-time dynamics of the prices.

First we conjecture that the quasi-static equilibrium of the dynamic price mechanism must satisfy the basic properties of the Walrasian equilibrium.

Conjecture 1. Consider the dynamic price mechanism under the quasi-static assumptions (i.e., $\tilde{\epsilon} = o(1)$), $\beta = \Theta(d)$, and a d-replicated economy. Assume that there is a quasi-static equilibrium set of prices such that for two VM classes, c and c', the prices satisfy $p_{c'} - p_c = \Omega(d^{-1/2})$. Then, any user who purchases $O(1)$ VMs from class c, purchases at most $o(d^{-1/2})$ VMs from class c' with high probability (i.e., with probability tending to 1 as $d \to \infty$; henceforth w.h.p.).

Heuristic Argument. The key idea is that once the user has sufficient budget to buy a VM from class c, then with high probability she will get the chance to do so before her budget increases sufficiently to be able to purchase a VM from class c'. Let $\Delta := p_{c'} - p_c$ and assume that user i has a budget between p_c and $p_c + \Delta/2$. Then the expected time until user i purchases a VM of class c is $O(1/(d\alpha))$; this uses the fact that there is a constant probability of a user being able to purchase a VM that they request, since typically there are only a constant number of users with sufficient budget to purchase a VM (if this were not the case, the price mechanism would raise the prices). However, assuming $\Delta = \Omega(d^{-1/2})$, the expected time until the budget of user i could reach $p_{c'}$ is $\Omega(1/(d^{1/2}\gamma s))$. Thus for d sufficiently large, user i can rarely afford VMs from the more expensive class c' and the ratio of purchases of VMs in class c to those in class c' is $o(d^{-1/2})$.

Conjecture 2. Under the quasi-static assumptions, $\beta = \Theta(d)$, and in a d-replicated economy, in the limit as $d \to \infty$, there exists a unique equilibrium of the dynamic price mechanism which is the same as the Walrasian equilibrium.

Heuristic Argument. Conjecture 1 shows that in the equilibrium the VMs and users will partition based on (almost) equal prices, i.e., each cluster of VMs has prices that are within $o(d^{-1/2})$ of each other w.h.p. and has a related subset of users for whom those VMs are the least expensive of their allowable VMs so they only buy goods from that subset of VM classes. Theorem 1 shows that the Walrasian equilibrium also must partition in this manner. The choice of $\beta = \Theta(d)$ guarantees efficiency: w.h.p. only $O(1)$ VMs are in the reserve. From this one can see that the two equilibria must be identical, and moreover the Walrasian equilibrium is known to exist in this setting.

Using this, we further conjecture that under the quasi-static assumptions the dynamic pricing mechanism will converge to the Walrasian equilibrium in large economies ($d \gg 1$).

Conjecture 3. Consider the dynamic price mechanism under the quasi-static assumptions detailed above (i.e., $\tilde{\epsilon} = o(1)$), and $\beta = \Theta(d)$. Let $p^{(d)}(t)$ denote the prices at time t for the d-replicated economy. For any fixed t, in the limit as $d \to \infty$, starting from any initial condition $p > 0$, the prices $p^{(d)}(t)$ will converge in probability to the Walrasian equilibrium.

Heuristic Argument. First, to simplify the notation assume that in the Walrasian equilibrium all the prices p^* are the same and that for every class c there exists another class c' such that some user purchases VMs from both classes. Now consider a set of prices $p > 0$ where the largest price p_c is unique and greater than p^*_c. Following the arguments in Conjecture 1, one can see that w.h.p. for sufficiently large d the only users who purchase VMs in class c are those for whom these are their only allowable machines. By assumption there exist other users who would also be purchasing these VMs in the Walrasian equilibrium. Thus, the demand for these VMs is lower under p then under $p*$ in the equilibrium which implies that the price adjustment process would cause this price to fall. Similarly, if there was a unique lowest price $p_{c'} < p^*_{c'}$ then a similar argument would show that that price would rise under the price adjustment process. Thus, assuming these unique highest and lowest prices, we see that the highest price (greater than the equilibrium price) would fall and the lowest price (smaller than the equilibrium price) would rise over time, leading to convergence at the equilibrium prices using a contraction mapping in the L_∞ norm.

To extend this to the case where prices are not necessarily unique, both in $p*$ and p, uses the same reasoning but requires a detailed analysis of the underlying bipartite graph. For example, suppose p partitions into two sets of VM classes, one with every price equal to q and the other with every price equal to q' with $q' > p^*$ and $q' > q$. Then, either there exists some user who purchased goods from VM classes in both subsets in the Walrasian equilibrium, in which case the higher price class of VMs will face a drop in price, as required in the proof, or there does not exist any such user and the two subsets do have different prices in the Walrasian equilibrium. In the latter case, one can apply the contraction mapping argument separately to each of the separate subsets of VM classes.

We believe that these heuristic arguments can be turned into rigorous proofs once multi-time-scale systems of this type are well understood.

We see from these arguments that the limiting behavior of user-based budget-constrained dynamic pricing is fair, while our example in Section 3.1 shows that they will be efficient.

4 Simulation Results

We performed detailed simulations on the effects of user-based budget-constrained dynamic pricing. Due to space constraints the details and figures will appear in the full version of the paper; here we present our main conclusions.

For one, large β is necessary for efficiency, but there is a tradeoff between fairness and efficiency. The choice of $\beta = \kappa m$ for an appropriately chosen small constant κ (e.g., $\kappa = 0.1$) combines both efficiency and fairness.

There is also a tradeoff between efficiency and price stability, which depends on the price update parameter ϵ. Small ϵ leads to stable prices, but this decreases efficiency, and a further drawback of it is that prices are slow to adapt to sudden changes in the environment.

Finally, as the system size grows, prices and allocations converge to their respective values in equilibrium, provided the parameters are chosen appropriately. However, finite size effects are strong, resulting in price differences for finite systems, which can be unfair.

5 Conclusions

Electronic markets used for allocating resources are becoming widespread. With the advent of cloud computing, it is clear that these markets are here to stay, and it is of crucial importance to understand and implement pricing mechanisms that reach certain stringent goals—in the case of data centers where pricing is an internal mechanism, these are efficiency and fairness.

We have introduced a user-based budget-constrained dynamic pricing mechanism that can realistically be applied to systems of interest, such as those arising in cloud computing. Its primary advantages include that it is *simple*, and *robust* in adapting to changes in the architecture of the underlying system.

Our main contribution is the study of the allocative properties and dynamic stability of this pricing mechanism. This is important due to issues arising from the fact that equilibrium prices are highly degenerate in systems of interest. We show that dynamic pricing solves this problem, by creating slight differences in prices and thus providing incentives to move away from over-demanded resources. As the size of the system grows, these differences in prices disappear, and the prices converge to the standard Walrasian equilibrium prices.

However, importantly, finite-size effects are strong, and the allocations resulting from price differences can be unfair. In addition, improper choices of price update parameters can lead to significant instabilities in prices, which could be problematic in real cloud computing systems, e.g., by allowing manipulations by users. We uncover key scaling rules for the parameters that reduce these instabilities, and which therefore can be of use to system administrators.

While we have restricted our analysis to the allocation of compatible virtual machines in this paper, we believe the extension of such budget-constrained dynamic pricing models to more general economies would be valuable. For example, in cloud computing the internal pricing of resources such as CPU, memory or data transfer, rather that just VMs, is an important issue [8], and extending our model to that setting would be useful.

Acknowledgments.

Will appear here in the deanonymized version.

References

1. G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica. Disk-locality in datacenter computing considered irrelevant. In *Proceedings of the 13th USENIX Workshop on Hot Topics in Operating Systems*, pages 1–5, 2011.
2. A.A. Bhattacharya, D. Culler, E. Friedman, A. Ghodsi, S. Shenker, and I. Stoica. Hierarchical Scheduling for Diverse Datacenter Workloads. In *Proceedings of ACM Symposium on Cloud Computing (SoCC'13)*, 2013.
3. B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.
4. R. Cole and L. Fleischer. Fast-Converging Tatonnement Algorithms for One-Time and Ongoing Market Problems. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 315–324. ACM, 2008.
5. G. Debreu and H. Scarf. A limit theorem on the core of an economy. *International Economic Review*, 4(3):235–246, 1963.
6. S. Dobzinski, R. Lavi, and N. Nisan. Multi-unit auctions with budget limits. *Games and Economic Behavior*, 74(2):486–503, 2012.
7. E. J. Friedman, J. Y. Halpern, and I. Kash. Efficiency and Nash Equilibria in a Scrip System for P2P Networks. In J. Feigenbaum, J. Chuang, and D. Pennock, editors, *ACM Conference on Electronic Commerce (EC)*, pages 140–149, 2006.
8. A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dominant resource fairness: fair allocation of multiple resource types. In *USENIX NSDI*, 2011.
9. A. Ghodsi, M. Zaharia, S. Shenker, and I. Stoica. Choosy: max-min fair sharing for datacenter jobs with constraints. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 365–378. ACM, 2013.
10. G. Goel, V. Mirrokni, and R. Paes Leme. Polyhedral Clinching Auctions and the Adwords Polytope. In *Proceedings of the 44th Symposium on Theory of Computing*, pages 107–122. ACM, 2012.
11. E.L. Hahne and R.G. Gallager. Round robin scheduling for fair flow control in data communication networks. *NASA STI/Recon Technical Report N*, 86:30047, 1986.
12. B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A.D. Joseph, R.H. Katz, S. Shenker, and I. Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of NSDI*, 2011.
13. A.M. Il'in, R.Z. Khasminskii, and G. Yin. Singularly Perturbed Switching Diffusions: Rapid Switchings and Fast Diffusions. *Journal of Optimization Theory and Applications*, 102(3):555–591, 1999.
14. J. Sweeney and R.J. Sweeney. Monetary theory and the great Capitol Hill Baby Sitting Co-op crisis: comment. *Journal of Money, Credit and Banking*, 9(1):86–89, 1977.
15. R.H. Timberlake. Private production of scrip-money in the isolated community. *Journal of Money, Credit and Banking*, 19(4):437–447, 1987.
16. H. Uzawa. Walras' Tatonnement in the Theory of Exchange. *The Review of Economic Studies*, 27(3):182–194, 1960.
17. L. Walras. *Elements of Pure Economics Or The Theory of Social Wealth*. George Allen & Unwin Ltd., 1954.
18. G. Yin and Q. Zhang. *Continuous-Time Markov Chains and Applications: A Two-Time-Scale Approach*, volume 37. Springer, 2012.