Game Based Capacity Allocation for Utility Computing Environments

Benjamin Yolken
yolken@stanford.edu

Nicholas Bambos
bambos@stanford.edu

Stanford University
Stanford, CA

ABSTRACT

Utility computing has the potential to greatly increase the efficiency of IT operations by sharing resources across multiple users. This sharing, however, introduces complex problems with regards to pricing and allocating these resources in a way that is fair, easy to implement, and economically efficient. In this paper, we study a queue-based model that attempts to address these issues. Each client / user has a continuous flow of jobs that need to be processed. The service rate each receives, however, is proportional to a *bid* it submits to the system operator. Assuming that user costs are some function of their average backlogs plus their bid amounts, we use this allocation mechanism to construct an economic *game*.

Much previous research has shown that these types of allocation games have desirable properties if the cost functions are well-defined and convex over the space of possible outcomes. Because of its queueing interface, however, our model induces functions that *do not* satisfy the latter, commonly assumed properties. In spite of these complications, we show that the game still has a unique equilibrium and that the system will converge to this point if users iteratively make "best response" updates to their bids. Finally, we discuss some numerical examples, exploring the rate of this convergence as well as some monotonicity properties of the resulting outcomes. Future research will expand this model to broader classes of service and also rigorously investigate its efficiency.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Management,Performance,Theory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Gamecomm 2008 October 20, 2008, Athens, GREECE
Copyright 2008 ICST 978-963-9799-31-8.

Keywords

Nash Equilibrium, queueing, utility computing

1. INTRODUCTION

Utility computing, loosely defined as the allocation and management of computer resources on a subscription basis, has received much recent attention in the IT world [13]. This excitement is well-founded: the latter concept has the potential to significantly reduce costs for clients, who essentially "rent" service on an as-need basis, while at the same time producing additional revenue for data center operators, whose resources are often underutilized. Working examples of such utility computing systems include Sun Microsystem's "Sun Grid" and Amazon's "Elastic Compute Cloud (EC2)"; other large players, such as Google and IBM, have announced plans to enter the business as well [11].

This new framework, however, introduces challenging problems with respect to resource allocation, quality-of-service (QOS), and pricing. Utility computing will be provided by large data centers with many thousands of individual components (CPUs, disk drives, switches, etc.); to provide service at a reasonable cost, these resources will need to be internally managed in an efficient, autonomous manner. These components, for instance, will need controls that automatically perform load balancing, fault detection, etc.

At the same time, these providers will also need an external *pricing and resource allocation* interface for the clients. The latter, which is the emphasis in the remainder of this paper, is necessary for setting resource prices, collecting job requirements and payments from these clients, and then allocating the resources to their jobs accordingly. Ideally, this pricing / allocation scheme (1) is computationally tractable, (2) has low signalling requirements, (3) leads to stable behavior over time, and (4) produces economically efficient (or at least "near-efficient") outcomes. The fourth property is often the hardest to satisfy; in particular, the commonly deployed "one size fits all" pricing scheme (like SunGrid's "$1 per CPU hour") is usually *not efficient* because it is not responsive to demand fluctuations or client heterogeneity.

In this paper, we propose one such model for pricing and allocating a utility computing resource. In particular, we assume that client tasks are represented as job flows in a controlled queueing system. These jobs arrive to the system through a fixed, random process, are stored in a buffer, and then are serviced by the resource in a first come, first served (FCFS) manner. The service rate, however, is set through an auction-like, proportional share mechanism- users submit bids to the system operator and then receive service propor-

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

tional to their bids and the bids of the other users. Clients, therefore, must balance the delay experienced by their jobs versus the added cost of buying additional resource capacity. Using ideas from game theory, we show that such a scheme has a unique *Nash Equilibrium* and moreover that this point can be reached in a distributed, asynchronous manner. We then prove that these equilibrium bids are responsive to user demands and delay requirements in an intuitive way. Thus, our scheme has the potential to be much more efficient than other, naive allocation approaches. We leave a formal proof of this, however, for future work.

Our approach here combines features from a number of different research threads. On the one hand, much previous work has studied game theory in the context of queueing systems [6]. Most of this work, however, has focused on strategic interactions at the *job* level; each player of the game corresponds to a single job which must decide, for example, whether it is willing to join the system at some announced price (e.g. [12]) or how much it is willing to bid for service priority (e.g. [5]). Very little work has looked at game theoretic queueing in the aggregate *flow* context used here.

On the other hand, a significant body of research has studied methods for allocating divisible computing / communication resources. Several papers, including [4], [7], and [10] have looked at this type of allocation problem from a theoretical economic perspective, proving existence, uniqueness, efficiency loss bounds, etc. under various assumptions. Others, such as [3], [8], and [2] have focused instead on the implementation side of the problem, studying specific system architectures, communication schemes, and convergence algorithms.

While our work is similar in spirit to many of the previous papers, our original contribution here is in applying these results specifically to queue-related valuation / utility functions. This queueing environment induces functional forms which do not conform to the ones traditionally used in the literature; the latter generally assume, for instance, that all utility and cost functions are continuous and well-defined over the space of possible outcomes. In contrast, our model leads to functions with asymptotes and infinite values, features that require us to derive equilibria properties using innovative and unique approaches.

The remainder of this paper is organized as follows. In Section 2, we describe our model and the associated resource allocation mechanism. In Section 3, we characterize the equilibria in our model, proving existence, uniqueness, and other properties. Section 4 discusses some numerical illustrations of our results. Finally, we conclude and give directions for future research in Section 5.

2. MODELS AND NOTATION

2.1 Resource Model

Consider a resource being shared in a utility computing system. This resource may be a literal physical resource (e.g., a server), or a "virtual" collection of such components that have been aggregated together for client use. Assume that this resource has a fixed processing rate, $\gamma > 0$, that can be arbitrarily split between clients, for instance through virtualization software.

Suppose that this resource is being used by N non-cooperating, selfish users / clients, indexed as $i = 1, 2, \ldots, N$. Associated with each client is a *fixed*, random flow of jobs that require service from the resource. For simplicity, we assume that (1) user i's flow is Poisson with rate $\lambda_i > 0$ and (2) each user i job has a service requirement drawn independently from an exponential distribution with parameter $\mu_i > 0$. Define the "load intensity" as $\rho_i = \frac{\lambda_i}{\mu_i}$. For notational clarity, we use λ, μ, and ρ in the sequel to represent the "vectorized" versions of the previous parameters.

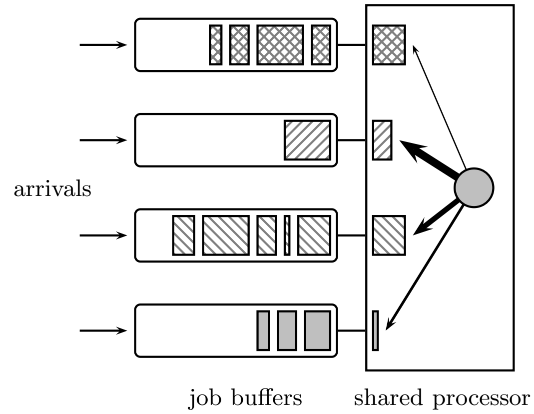


Figure 1: Illustration of model for shared resource. Client jobs arrive and are buffered while awaiting service. Each head-of-line job receives processing at a client-specific rate.

Let θ_i represent the fraction of the shared resource devoted to servicing user i's jobs, with $\sum_i \theta_i \leq 1$. Assuming that all rates are measured in common units, it follows from basic queueing theory that this user experiences an average job backlog given by

$$B(\theta_i) = \begin{cases} \frac{\rho_i}{\gamma\theta_i - \rho_i} & \text{if } \theta_i > \frac{\rho_i}{\gamma} \\ +\infty & \text{otherwise} \end{cases} \tag{1}$$

Let $D(\theta_i)$ represent user i's average job delay. By Little's Law [9], the previous two quantities are related as:

$$B(\theta_i) = \lambda_i D(\theta_i) \tag{2}$$

Hence, the average backlog is equivalent to a "rate weighted" average delay.

Note that infinite backlogs (and delays) are possible if user i's share, θ_i, is not "big enough." For mathematical tractability, we assume that there exists some allocation such that $B(\theta_i) < \infty$ for all i. One can show that this is equivalent to the condition

$$\sum_i \rho_i \leq \gamma \tag{3}$$

If the latter condition does not hold then the system is overloaded, and one or more users need to be excluded. For lack of space, we ignore this case here and leave the issue as a topic for future study.

2.2 Incentive Model

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

We now consider the resulting allocation problem. If the system operator had complete information about the users (i.e., their job rates, their delay sensitivities, etc.), then this authority could distribute the resource in some kind of "optimal" way. Unfortunately, however, this all-knowing assumption seems unreasonable. Moreover, absent some additional structure, the users have no incentive to reveal their true costs if asked. These agents, for instance, might artificially inflate their delay sensitivities to obtain a greater-than-fair share of processing power from the resource.

As done in previous work, we propose an auction-like *mechanism* to solve this problem. In particular, users submit bids to a central authority; each then receives a resource share in proportion to its bid and the bids of its competitors. Ideally, those users who value delay more will submit higher bids and receive comparatively more service. In addition, these bids should reflect the congestion faced by the system- as γ decreases, for instance, we would expect bids to increase. Thus, this auction-like system has the potential to allocate resources in a "meaningful" way without requiring total knowledge or truthful cost revelation.

We formalize this framework as follows. Suppose that each user submits some nonnegative bid, $w_i \geq 0$. Let $\mathbf{w}$ and $\mathbf{w}_{-i}$ represent, respectively, the vector of all bids and the vector of bids of all users other than i. Given this information, the system then allocates the resource processor as:

$$\theta_i(\mathbf{w}) = \begin{cases} \gamma \frac{w_i}{\sum_i w_i} & \text{if } \sum_i w_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In other words, the processing rate for each client is *proportional* to this client's share of the total bids. For this reason, this type of allocation scheme is often referred to as a *proportional share* mechanism [2].

Given this allocation rule, we assume that each user then experiences a cost given by:

$$C_i(\mathbf{w}) = v_i B_i(\theta_i(\mathbf{w})) + c_i w_i \quad (5)$$

for strictly positive constants v_i and c_i. The former encode the user's delay cost sensitivity at each queue; the latter, on the other hand, represent user k's "bid cost". Assuming that all user bids are in some common unit (e.g., dollars), we can, without loss of generality, set $c_i = 1$ for $i = 1 \ldots N$.

Given fixed bids for all other users, it follows that user k can decrease the first, "delay" term by increasing its bid. This move, on the other hand, will increase the second, "bid cost" term. Thus, our cost function captures the tradeoffs between *delay* on the one hand and *money* on the other. As discussed in the introduction, such a setup has the capability of providing "meaningful" payments with minimal central authority knowledge.

In such settings, it is natural to model the interactions between users as an economic *game*. In particular, we assume that all users announce their bids simultaneously and experience the costs described above.[1] As is commonly done in the literature, we restrict our attention to outcomes which are *Nash Equilibria* in pure strategies, in other words bid vectors $\mathbf{w}$ satisfying:

[1]Games are usually described in terms of payoffs or utilities, which users seek to maximize, rather than costs. We use the latter here for ease of notation. One can convert to the more standard form by multiplying our functions by -1.

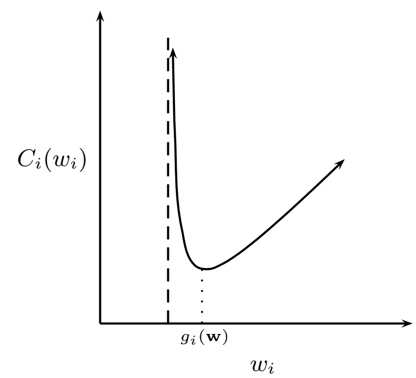


Figure 2: Total cost incurred by user i as a function of its own bid, w_i, holding all other bids fixed; i's best response, $g_i(\mathbf{w})$, is the minimizer of this function.

$$C_i(w_i, \mathbf{w}_{-i}) \leq C_i(\bar{w}_i, \mathbf{w}_{-i}) \quad \forall \bar{w}_i \in \mathbb{R}_+ \quad (6)$$

for all users i. Such outcomes represent "stable points" at which no user has an incentive to deviate unilaterally.

Alternatively, one can define a Nash Equilibrium in terms of a "best response" function. To this end, let:

$$g_i(\mathbf{w}) = \arg\min_{w_i \in \mathbb{R}_+} C_i(w_i, \mathbf{w}_{-i}) \quad (7)$$

with $g(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}), \ldots, g_N(\mathbf{w})))$. It then follows that a bid vector, $\mathbf{w}$, is a Nash Equilibrium if $g(\mathbf{w})$ is well defined and

$$\mathbf{w} = g(\mathbf{w}) \quad (8)$$

i.e. $\mathbf{w}$ is a *fixed point* of this function. On the other hand, the best responses may not be well defined (e.g., if $\mathbf{w} = \mathbf{0}$). As we show in the next section, however, such cases cannot be equilibria. In our game, therefore, the former assertion applies in both directions.

3. EQUILIBRIA PROPERTIES

3.1 Optimality Conditions

Consider a utility computing resource shared under the mechanism above. With some abuse of notation, let $w_{-i} = \sum_{j \neq i} w_j$, i.e., the sum of all bids of all users other than i. Combining the cost and allocation equations from the previous section, we then have

$$C_i(\mathbf{w}) = \begin{cases} \frac{v_i \rho_i}{\gamma \frac{w_i}{w_i + w_{-i}} - \rho_i} + w_i & \text{if } \gamma \frac{w_i}{w_i + w_{-i}} > \rho_i \\ +\infty & \text{otherwise} \end{cases} \quad (9)$$

Note that $w_i = 0$ always results in infinite costs for player i, even if all other bids are also 0. Also note that, for fixed $\mathbf{w}_{-i}$, $C_i(\cdot)$ has an asymptote at $w_i = w_{-i} \frac{\rho_i}{\gamma - \rho_i}$. Hence, standard game theory tools that require continuity (e.g., Rosen's theorem) cannot be easily applied.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

After some algebra, we find that the best response functions are given by:

$$g_i(\mathbf{w}) = \frac{\rho_i w_{-i} + \sqrt{v_i \rho_i \gamma w_{-i}}}{\gamma - \rho_i} \quad (10)$$

provided that $w_{-i} > 0$. If $w_{-i} = 0$, then the best response is not well-defined; any $w_i > 0$ will strictly reduce this player's cost. However, as shown later, this is a condition that cannot happen in equilibrium.

3.2 Equilibrium Existence and Uniqueness

Note that the best response functions satisfy the following two properties for all $\mathbf{w} > \mathbf{0}$:

1. $\mathbf{w} \geq \bar{\mathbf{w}} \ \Rightarrow \ g(\mathbf{w}) \geq g(\bar{\mathbf{w}})$
2. $\alpha g(\mathbf{w}) > g(\alpha \mathbf{w}) \qquad \forall \alpha > 1$

Yates [15] refers to these as *monotonicity* and *scalability*, respectively, and shows that they imply the existence and uniqueness of a fixed point. His theorems, however, require that $g(\mathbf{0})$ is well-defined and positive, something clearly absent here. This difference necessitates some novel approaches, as we no longer have an easy "starting point" for defining monotone best response sequences. We describe these in the following theorems and proofs.

THEOREM 1. *There exists at least one Nash Equilibrium for the given resource allocation game.*

PROOF. Let $w_i = \rho_i$ for all i. Note that in this case

$$\begin{aligned} \gamma \tfrac{w_i}{w_i + w_{-i}} &= \gamma \tfrac{\rho_i}{\sum_i \rho_i} \\ &= \rho_i \tfrac{\gamma}{\sum_i \rho_i} \\ &> \rho_i \end{aligned} \quad (11)$$

Hence, all costs are finite and all best responses are well-defined. We now seek some positive scalar, β, such that:

$$\begin{aligned} & \beta \mathbf{w} > g(\beta \mathbf{w}) \\ \iff \ & \beta \rho_i > \beta \rho_i \left[\tfrac{\sum_{j \neq i} \rho_j}{\gamma - \rho_i} \right] + \tfrac{\sqrt{\beta v_i \rho_i \gamma \sum_{j \neq i} \rho_j}}{\gamma - \rho_i} \qquad \forall i \end{aligned} \quad (12)$$

Since

$$\frac{\sum_{j \neq i} \rho_j}{\gamma - \rho_i} < 1 \quad (13)$$

it follows that the left hand sides of (12) will all dominate the right hand sides for β large enough. Let β^+ represent one such value of this constant satisfying these conditions.

Likewise, note that for β small enough, the reverse will happen: $\beta \rho < g(\beta \rho)$. Let β^- represent one such value satisfying the latter condition.

Consider now applying the following iterative procedure. Set $\mathbf{w}(0) = \beta^+ \rho$ and let $\mathbf{w}(1) = g(\mathbf{w}(0))$, $\mathbf{w}(2) = g(\mathbf{w}(1))$, $\ldots$, $\mathbf{w}(n) = g(\mathbf{w}(n-1))$. From (12) and the discussion above, we have $\mathbf{w}(0) \geq \mathbf{w}(1)$. Applying the best response function to both sides and using the monotonicity of the $g(\cdot)$ function, we get:

$$\begin{aligned} g(\mathbf{w}(0)) &\geq g(\mathbf{w}(1)) \\ \iff \quad \mathbf{w}(1) &\geq \mathbf{w}(2) \end{aligned} \quad (14)$$

Iteratively continuing this procedure, we get that $\mathbf{w}(n-1) \geq \mathbf{w}(n)$ for all integers $n \geq 1$. Hence it follows that $\mathbf{w}(n)$ is monotonically decreasing (componentwise) in n.

By a similar argument, if we define $\hat{\mathbf{w}}(0) = \beta^- \rho$, we have that the latter is a monotonically increasing sequence in n. Note that $\mathbf{w}(n) > \mathbf{0}$ and $\hat{\mathbf{w}}(n) > \mathbf{0}$ for all n, implying that the best responses are well-defined at each sequence step. Moreover, we have $\beta^- \rho < \beta^+ \rho$. So, by the monotonicity property

$$\hat{\mathbf{w}}(n) \leq \mathbf{w}(n) \quad (15)$$

as well. This, in turn, implies that each sequence converges to a (positive) point. Let $\bar{\mathbf{w}}$ represent this limit for the right hand, decreasing sequence. Since $\bar{\mathbf{w}} > \beta^- \rho > \mathbf{0}$, it follows that the $g(\cdot)$ functions are well defined and continuous in a sufficiently small neighborhood around this point. We therefore have:

$$g(\bar{\mathbf{w}}) = \bar{\mathbf{w}} \quad (16)$$

i.e., $\bar{\mathbf{w}}$ is a Nash Equilibrium for the game, as required. □

THEOREM 2. *The resource allocation game has a unique Nash Equilibrium.*

PROOF. Suppose by contradiction that we have two Nash Equilibria, $\mathbf{w}$ and $\hat{\mathbf{w}}$. We first claim that both of these must have all components positive. For otherwise, we either have $\mathbf{w} = \mathbf{0}$, in which case every user has an incentive to deviate or $w_i = 0, w_{-i} > 0$ for some i, in which case the latter user can strictly decrease its cost by changing its bid to $g_i(\mathbf{w}) > 0$. This implies that both points must also be fixed points of $g(\cdot)$.

Arguing analogously to [15], we can assume without loss of generality that $w_i < \hat{w}_i$ for some i and scale $\mathbf{w}$ by α so that $\alpha \mathbf{w} \geq \hat{\mathbf{w}}$ with $\alpha w_i = \hat{w}_i$. We then have

$$\begin{aligned} \hat{w}_i &= g_i(\hat{\mathbf{w}}) \\ &\leq g_i(\alpha \mathbf{w}) \\ &< \alpha g_i(\mathbf{w}) \\ &= \alpha w_i \end{aligned} \quad (17)$$

a clear contradiction. Thus, we must have $\mathbf{w} = \hat{\mathbf{w}}$, as claimed. □

3.3 Convergence

Given a game like the one described here, a natural question to ask is how we can *compute* a Nash Equilibrium that we know exists. Indeed, if this process is overly complex, then the existence and uniqueness analysis done above has little value in real-world applications.

Ideally, this computation is also possible in a *distributed* fashion. In particular, each user updates its strategy at various points in time based on the currently observed choices of the other users. In this way, the system converges to a Nash Equilibrium point without any significant intervention required by a central authority / administrator.

As discussed in [16], one particularly natural procedure is for each user to take a *best response* at each such update point. In other words, given some current set of bids $\mathbf{w}$, user i modifies its bid by setting $w_i = g_i(\mathbf{w})$. To do this accurately, each user needs only the appropriate value of w_{-i} (or $\sum_i w_i$), not the individual bids of the other users.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

We say that this updating is *synchronous* if all users apply the latter at each update point. This leads to the following algorithm:

Algorithm 1 Synchronous Best Response Dynamics (SBRD)

Given $\mathbf{w}(0) > \mathbf{0}$
Set $t \leftarrow 0$
repeat
  $\mathbf{w}(t+1) = g(\mathbf{w}(t))$
  $t \leftarrow t+1$
until converged

The exact convergence condition is a matter of "preference." One possibility is to terminate the algorithm when $||\mathbf{w} - g(\mathbf{w})||_\infty < \epsilon$ for some small $\epsilon > 0$.

We now show that the previous algorithm converges to the (unique) Nash Equilibrium:

THEOREM 3. *The SBRD algorithm converges to the unique NE from any positive starting point,* $\mathbf{w}(0) > \mathbf{0}$.

PROOF. Let β^+ and β^- be positive constants with the properties discussed in the proof of Theorem 1 above. Since $\mathbf{w}(0) > \mathbf{0}$, it follows that there exists a β^- small enough and a β^+ large enough such that

$$\beta^- \rho \leq \mathbf{w}(0) \leq \beta^+ \rho \tag{18}$$

We now consider applying the best response function to each term in the above inequality. Let $[\cdot](n)$ represent the result of applying $g(\cdot)$ to the bracketed vector n times. By monotonicity, we necessarily have

$$[\beta^- \rho](n) \leq \mathbf{w}(n) \leq [\beta^+ \rho](n) \tag{19}$$

for all n. From the discussion in Theorem 1 above, $[\beta^+ \rho](n)$ is a monotonically decreasing sequence converging to $\bar{\mathbf{w}}$, the (unique) fixed point of $g(\cdot)$. In addition, $[\beta^- \rho](n)$ is a monotonically increasing sequence, necessarily converging to this same, unique fixed point.

Thus, $\mathbf{w}(n)$ is "pinched" between two sequences, both converging to the same point. It follows that the latter sequence also converges to this point, which gives us the desired result. □

Another, less stringent convergence algorithm involves taking these updates *asynchronously*. More formally, assume that all possible update times are indexed as $t = 0, 1, 2, \ldots$, and let T^i represent the set of these times at which user i updates w_i. Assume that this set is infinite so that updates are taken infinitely often for each user. As in [16], this gives us the following algorithm:

Algorithm 2 Asynchronous Best Response Dynamics (ABRD)

Given $\mathbf{w}(0) > 0$
Set $t \leftarrow 0$
repeat
  for $i = 1 \ldots N$ **do**
    if $t \in T^i$ **then**
      $g_i(t+1) = g_i(\mathbf{w}(t))$
    else
      $g_i(t+1) = w_i(t)$
    end if
  end for
  $t \leftarrow t+1$
until converged

As in the synchronous case, we consider the algorithm converged when $||\mathbf{w} - g(\mathbf{w})||_\infty < \epsilon$ for some small $\epsilon > 0$. We then have the following theorem:

THEOREM 4. *The ABRD algorithm also converges to the unique NE from any positive starting point,* $\mathbf{w}(0) > \mathbf{0}$.

PROOF. Take $\mathbf{W}(n) = \{\mathbf{w} \mid \beta^- \rho(n) \leq \mathbf{w} \leq \beta^+ \rho(n)\}$. We can then apply the "Asynchronous Convergence Theorem" from [1] analogously to [15], so we omit the details here, given the limited space. □

3.4 Revenue, Price, and Share Monotonicity

Since we interpret the w_i values as the bids paid by each client for use of the resource, it makes sense to interpret the value $\sum_i w_i$ as the total *revenue* obtained by the operator for providing γ service capacity to these user jobs. It seems intuitive then that the former value should increase in equilibrium as the system gets increasingly congested, i.e. as one or more ρ_i values increase. We would also expect that this revenue is increasing in v_i; as users become increasingly delay sensitive, they are willing to pay more for service.

Both of these hypotheses are provably true as we discuss in the following theorem:

THEOREM 5. $\sum_i w_i$ *is increasing in* ρ_i *and* v_i *for any player,* i

PROOF. Let $\mathbf{w}(0)$ represent the unique equilibrium under the arrival intensity vector $\rho = (\rho_1, \rho_2, \ldots, \rho_i, \ldots)$. Now consider the same game with the arrival intensities given by $\bar{\rho} = (\rho_1, \rho_2, \ldots, \bar{\rho}_i, \ldots)$ where $\bar{\rho}_i > \rho_i$ but still $\sum_i \rho_i < \gamma$.

Consider two best response functions:

1. $g^1(\mathbf{w}) = (w_1, w_2, \ldots, g_i(\mathbf{w}), \ldots)$ and
2. $g^2(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}), \ldots, w_i, \ldots)$

where the *best response in each case uses the modified arrival intensities* $\bar{\rho}$, as opposed to the original ones ρ. Let

$$\mathbf{w}(n) = \begin{cases} g^1(\mathbf{w}(n-1)) & \text{if } n = 1, 3, 5, \ldots \\ g^2(\mathbf{w}(n-1)) & \text{if } n = 2, 4, 6, \ldots \end{cases} \tag{20}$$

In other words, we alternate between taking the first- and second-type best responses. Another equivalent point of view is that there are *two phases* of a best response process.

Since $g_i(\mathbf{w})$ is increasing in ρ_i for fixed $\mathbf{w}_{-i}$, we necessarily have

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

$$\begin{array}{rcl} \mathbf{w}(0) & \leq & g^1(\mathbf{w}(0)) \\ & = & \mathbf{w}(1) \end{array} \tag{21}$$

By the monotonicity of the individual best responses, we then get

$$\begin{array}{rrcl} & g^2(\mathbf{w}(0)) & \leq & g^2(\mathbf{w}(1)) \\ \Longrightarrow & \mathbf{w}(0) & \leq & \mathbf{w}(2) \end{array} \tag{22}$$

since $g^2(\mathbf{w}(0)) = \mathbf{w}(0)$ by the assumption that this point is a Nash Equilibrium under ρ.

Now suppose that

$$\mathbf{w}(0) \leq \mathbf{w}(n) \tag{23}$$

for some even $n > 1$. By the monotonicity of $g_i(\mathbf{w})$ in both $\mathbf{w}$ and ρ_i, we then necessarily have that

$$\mathbf{w}(0) \leq g^1(\mathbf{w}(0)) \leq g^1(\mathbf{w}(n)) \tag{24}$$

implying

$$\mathbf{w}(0) \leq \mathbf{w}(n+1) \tag{25}$$

where $n+1$ is odd. Taking a type-two best response to each side yields

$$\mathbf{w}(0) \leq \mathbf{w}(n+2) \tag{26}$$

By induction, it thus follows that $\mathbf{w}(0) \leq \mathbf{w}(n)$ for all $n > 0$. Furthermore, we note that the update procedure used here is a valid form of ABRD. By Theorem 4 above then, this update procedure must converge to some point, say $\bar{\mathbf{w}}$, with $\mathbf{w}(0) \leq \bar{\mathbf{w}}$. All user weights are greater in the new equilibrium, and therefore $\sum_i \bar{w}_i$ increases as well. This gives us the desired result.

The proof for v_i follows from the exact same steps (just replacing all ρ's by the corresponding v's). The only difference is that there is no restriction on $\sum_i v_i$, so $\bar{v}_i - v_i$ can be arbitrarily large. We omit the other details for brevity. □

If the quantity $\sum_i w_i$ is the total system revenue, then it makes sense to think of the quantity $\frac{\sum_i w^i}{\gamma}$ as the resulting *price* per unit of service capacity. By the theorem above, we have that this value is also increasing in ρ_i and r_i. Hence, the market price for service is set according to the user demands, and each user pays the same, "fair" amount per unit of capacity.

The previous theorem considers a property of the system as a whole. Using these results, we can then make the following claim about the individual equilibrium shares (i.e., θ values) of the players:

THEOREM 6. *θ_i is increasing in ρ_i and v_i for any player, i; all other user shares are decreasing in these quantities.*

PROOF. Consider the same setup and notation as in Theorem 5 above. As shown in the latter proof, an increase in ρ_i causes all components of the perturbed equilibrium, $\bar{\mathbf{w}}$ to increase. Therefore, for each player j, $\bar{w}_{-j}$, the perturbed "other bids sum" increases as well.

For each $j \neq i$, we thus have:

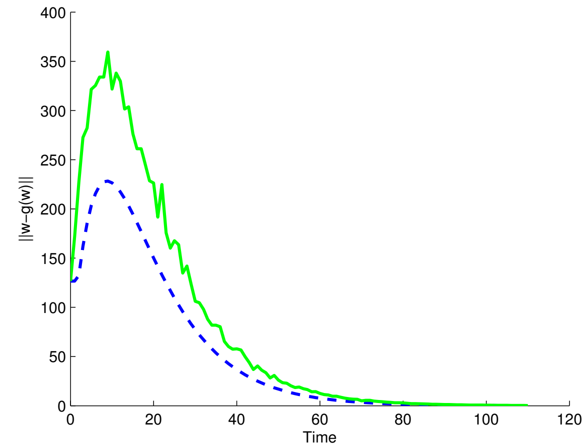


Figure 3: Convergence of SBRD (dotted blue line) and ABRD (solid green line) for instance of model with 1000 users and randomly generated parameters.

$$\begin{array}{rcl} \bar{\theta}_j & = & \frac{g_j(\bar{\mathbf{w}})}{g_j(\bar{\mathbf{w}})+\bar{w}_{-j}} \\ & = & \frac{\rho_j \bar{w}_{-j}+\sqrt{v_i\rho_i\gamma\bar{w}_{-j}}}{\gamma\bar{w}_{-j}+\sqrt{v_i\rho_i\gamma\bar{w}_{-j}}} \\ & < & \frac{\rho_j w_{-j}+\sqrt{v_i\rho_i\gamma w_{-j}}}{\gamma w_{-j}+\sqrt{v_i\rho_i\gamma w_{-j}}} \\ & = & \theta_j \end{array} \tag{27}$$

Since the mechanism allocations satisfy $\sum_j \theta_j = 1$, it follows that $\bar{\theta}_i > \theta_i$, as claimed.

The proof for the v_i case is similar and is omitted for brevity. □

This result, like the revenue one, makes intuitive sense: as we increase the "arrival intensity" or delay valuation for some user, this user's equilibrium share increases, at the expense of all the other users.

4. EXAMPLES AND SIMULATIONS

In this section, we briefly discuss two numerical examples of our model, showing that our proposed algorithms perform well and exploring the sensitivity of the resulting equilibria with respect to various parameters.

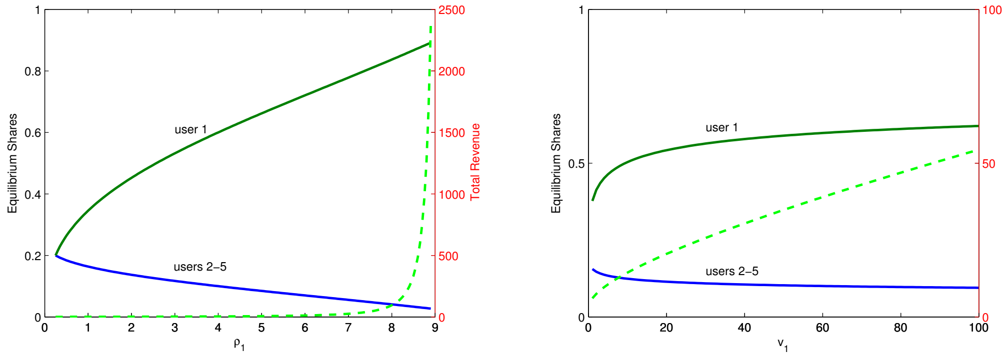
4.1 Example 1: Large System Convergence

To test the correctness of our algorithms, we created a large instance of our model with $N = 1000$ players. The values of each's client's v_i and ρ_i parameters were chosen uniformly at random from the the intervals $[1, 10]$ and $[0, 1]$, respectively. The total resource capacity was set to $\gamma = 600$, and care was taken (through multiple trials) to ensure that $\sum_i \rho_i < \gamma$ as required for system stability.

The system was started at a random point, $\mathbf{w}(0)$ (each component of which was chosen uniformly at random from $[0, 100]$). First, we ran SBRD. Then, from the same starting point, we ran a version of ABRD in which one client was chosen at random to perform an update at each time slot.

Figure 3 above shows the quantity $||\mathbf{w} - g(\mathbf{w})||_2$ as a function of time. For the basis of comparison, the time slots in

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415



(a) Sensitivity to ρ_1 (b) Sensitivity to v_1

Figure 4: User shares (solid lines) and total system revenue (dotted lines) as a function of ρ_1 and v_1. As the latter are increased, user 1's share increases at the expense of the others; system revenue also increases.

ABRD have been normalized so that the same number of updates occur in each time interval. We see that, after an initial "jump" as users increase their bids, both algorithms converge towards the fixed point at a reasonable rate. ABRD, however, does this in a slower and less smooth fashion because of its random nature; one user can be selected multiple times before another user, who may be extremely "unhappy" with its current bid, is allowed to make an update.

4.2 Example 2: Small System Sensitivity Analysis

To test the sensitivity of the equilibria points in our model to various parameter changes, we constructed another instance of our system, this time with 5 users. The first user's ρ_1 and v_1 parameters were varied while the other four were fixed to have $\rho_i = 0.25$, $v_i = 1$. We then evaluated the system equilibrium point, $\mathbf{w}$, for various values of ρ_1, v_1, and γ (the total system capacity). In particular, we performed two sets of runs: one varying $\rho_1 \in [0.25, 8.95]$, while holding $v_1 = 1$ and $\gamma = 10$ and a second set varying $v_1 \in [1, 100]$, while holding $\rho_1 = 0.25$ and $\gamma = 4$.

Figure 4(a) above shows the equilibrium shares of user 1 and users 2-5 as ρ_1 is varied in the first run set. This figure also plots the total system revenue versus this parameter. We see that, as ρ_1 is increased, this user's share of the resource increases at the expense of the other four users (as predicted by Theorem 6). At the same time, total system revenue increases in an exponential-like shape; as ρ_1 gets closer to 9, the system utilization approaches its upper limit, $\gamma = 10$. Average user backlogs get very large, and these players are forced into a "bidding war" to keep their service rates above ρ.

Figure 4(a) shows the same quantities, but this time as v_1 is varied in the second run set. Note that user 1's share increases significantly at first, but then saturates. Total revenue, on the other hand, increases in a nearly linear way; in contrast to the previous run set, there is no upper limit / capacity on the parameter v_1. Increasing this does give more of the resource to this user, but does not "stress" the system as much as a change in ρ.

5. CONCLUSION

In this paper, we have thus created a theoretical model for the pricing and sharing of resources in a utility computing system. Users submit bids for service and then pay a cost based on these bids and the average delays they observe. This scheme produces a unique Nash Equilibrium outcome, which is reached if users asynchronously "best respond" to their environment. Preliminary simulation results indicate that this convergence occurs at a relatively fast rate. Moreover, the resulting outcome responds intuitively with respect to changes in the system parameters.

Future research will attempt to extend our results here in a number of ways. First, we believe that strong existence, uniqueness, and convergence results may apply under more general assumptions on the client job flows and service requirements. Second, using ideas from queueing theory, we can adapt our model to the case that jobs are processed by multiple resources (i.e., flow in a multi-server network). Finally, our initial research suggests that it may be possible to bound efficiency losses under the proposed allocation scheme. Much exciting work remains to be done in analyzing models of utility computing.

6. REFERENCES

[1] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation*. Prentice Hall, 1989.

[2] P. Bodenbenner, J. Stößer, and D. Neumann. A pay-as-bid mechanism for pricing utility computing. In *Bled eConference*, June 2007.

[3] B. Chun and D. Culler. Market-based proportional resource sharing for clusters. Technical report, UC Berkeley, 2000.

[4] A. Dimakis, R. Jain, and J. Walrand. Mechanisms for efficient allocation in divisible capacity networks. In *IEEE Conference on Decision and Control*, Dec. 2006.

[5] R. Hassin. Decentralized regulation of a queue. *Management Science*, 41(1):163–173, 1995.

[6] R. Hassin and M. Haviv. *To Queue or Not to Queue: Equilibrium Behavior in Queueing Systems*. Springer, 2003.

[7] R. Johari and J. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathemtics of Operations Research*, 29(3):407–435, 2004.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415

[8] K. Lai, B. Huberman, and L. Fine. Tycoon: a distributed market-based resource allocation system. Technical report, HP Labs, 2008.

[9] J. Little. A proof of the queueing formula $l = \lambda w$. *Operations Research*, 9(3):383–387, May 1961.

[10] R. Maheswaran and T. Başar. Auctions for divisible resources: price functions, Nash Equilibrium, and decentralized update schemes. *Lecture Notes in Computer Science*, 2531:335–360, 2002.

[11] P. McDougall. Google, IBM join forces to offer cloud computing services. *InformationWeek*, May 2008.

[12] P. Naor. The regulation of queue size by levying tolls. *Econometrica*, 37(1):15–24, 1969.

[13] M. Rappa. The utility business model and the future of computing services. *IBM Systems Journal*, 43(1):32–42, 2004.

[14] S. Sanghavi and B. Hajek. Optimal allocation of a divisible good to strategic buyers. In *IEEE Conference on Decision and Control*, Dec. 2004.

[15] R. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13(7):1341–1348, 1995.

[16] B. Yolken, A. Miura-Ko, and N. Bambos. Incentive-compatible payments for service in product-form queueing networks. Working paper, Jan. 2008.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4415
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4415