

# A Brokering Framework for Large-Scale Heterogeneous Systems

Xin Bai<sup>1</sup>, Ladislau Bölöni<sup>1</sup>, Dan C. Marinescu<sup>1</sup>,  
Howard Jay Siegel<sup>2</sup>, Rose A. Daley<sup>3</sup>, and I-Jeng Wang<sup>3</sup>

<sup>1</sup>University of Central Florida  
School of Electrical Engineering  
and Computer Science  
Orlando, FL 32816-2362, USA  
{xbai,lboloni,dcm}@cs.ucf.edu

<sup>2</sup>Colorado State University  
Department of Electrical and Computer Engineering  
and Department of Computer Science  
Fort Collins, CO 80523-1373, USA  
HJ@colostate.edu

<sup>3</sup>Johns Hopkins University  
Applied Physics Laboratory  
11100 Johns Hopkins Road Laurel, MD 20723-6099, USA  
{Rose.Daley,I-Jeng.Wang}@jhuapl.edu

## Abstract

*In this paper we discuss the role of a broker in a market-oriented resource allocation model for large-scale heterogeneous systems. The simplified model is based upon a three party system, provider-broker-consumer. The allocation of resources is determined by their price, their utility to the consumer, and by the satisfaction of the consumer. The role of the broker is to add societal objectives to resource allocation algorithms and to mediate between greedy consumers and selfish providers. A simulation experiment was conducted to study the transient and the steady-state behavior of several performance measures, including the average consumer satisfaction, the average utility, and the hourly revenue.*

## 1 Introduction

Resource management in a large-scale heterogeneous system poses serious challenges due to the scale of the system, the heterogeneity and inherent autonomy of resource providers, and the large number of consumers and the diversity of their needs. Individual members of the community contribute computing cycles, storage, services, and communication bandwidth to the pool of resources available to the entire community. An efficient and fair utilization of the resources can be ob-

tained only through a scheme that gives incentives to the providers to share their resources and that encourages the consumers to maximize the utility of the received resources. A well-tested model for such a scheme is based on an economic model, in which the resources need to be paid for in a real or virtual currency. This model has the advantage of being provably scalable, and we can successfully reuse or adapt the models that govern the economy in our society.

Economic models are attractive for resource providers, beneficial for the consumers of resources, and have societal benefits. Indeed, providers benefit from contributing their resources and are encouraged to re-invest some of their profits into additional resources. Consumers enjoy fair treatment as the resource allocation is governed by rules that do not depend on the individual consumer. Moreover, providers and consumers have a say in the market and can make their own decisions to maximize their utility and/or profits. When system-centric scheduling policies are replaced by consumer-centric policies the system becomes more responsive to consumer needs and important problems are solved with higher priority. Economic models allow resource allocation and management to be more efficient, the demand and supply is regulated through economic activities and fewer resources are wasted, and excess capacity and overloading are averaged over a very large number of providers and consumers. Resources, e.g., CPU cycles, main memory, secondary storage, and network bandwidth/latency, are treated uniformly and

this can facilitate the design of large-scale distributed systems, such as computational grids. The system is more scalable and decision-making is distributed.

In an economic model, all the participants are considered self-interested. The resource providers are trying to maximize their revenues. The consumers want to obtain the maximum possible resources for the minimum possible cost. The large number of participants makes one-to-one negotiations expensive and unproductive. In this case, direct negotiation between resource providers and consumers is very inefficient. We need a broker to mediate access to resources from different providers. A broker is able to reconcile the selfish objectives of individual resource providers who want to maximize their revenues, with the selfish objectives of individual consumers who want to get the most possible utility at the lowest possible cost, and with some global, societal objectives, e.g., to maximize the utility of the system.

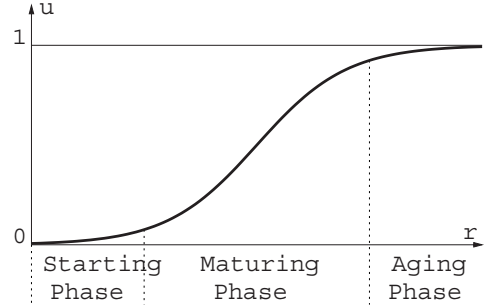
To formalize the objectives of the participants, we use: (i) a consumer *utility function*,  $0 \leq u(r) \leq 1$ , to represent the utility provided to an individual consumer, where  $r$  represents the amount of allocated resource; (ii) a provider *price function*,  $p(r)$ , imposed by a resource provider, and (iii) a consumer *satisfaction function*,  $s(u(r), p(r))$ ,  $0 \leq s \leq 1$ , to quantify the level of satisfaction; the satisfaction depends on both the provided utility and the paid price.

The utility function should be a non-decreasing function of  $r$ , i.e., we assume that the more resources are allocated to the consumer, the higher the consumer utility is. However, when enough resources have been allocated to the consumer, i.e., some threshold is reached, an increase of allocated resources would bring no improvement on the utility. For example, if a parallel application could use at most 100 nodes of a cluster, its utility reflected by a utility function does not increase if its allocation increases from 100 to 110 nodes. The above requirements are reflected by the following equations:

$$\frac{du(r)}{dr} \geq 0, \quad \lim_{r \rightarrow \infty} \frac{du(r)}{dr} = 0 \quad (1)$$

Sigmoid functions follow Equation (1) and are often used to model utility. A sigmoid is a tilted S-shaped curve that could be used to represent the life-cycles of living, as well as man-made, social, or economical systems. It has three distinct phases: an incipient or starting phase, a maturing phase, and a declining or aging phase, as shown in Figure 1.

In the context of resource allocation, a sigmoid quantifies the utility provided to an individual when the amount of resources allocated to the consumer in-



**Figure 1. A sigmoid includes three phases: the starting phase, the maturing phase, and the aging phase. Normally consumers do not want the amount of allocated resource to be at the starting phase because the utility is too low; they also do not want the amount of allocated resource to be at the aging phase because they can get a little lower utility while saving a large amount of resources.**

creases. We expect the utility to be a concave function and reaches saturation as the consumer gets all the resources it can use effectively. The consumer utility function could be a sigmoid [23]

$$u = u(r) = \frac{(r/\omega)^\zeta}{1 + (r/\omega)^\zeta} \quad (2)$$

where  $\zeta$  and  $\omega$  are constants provided by the consumer,  $\zeta \geq 2$ , and  $\omega > 0$ . Clearly,  $0 \leq u(r) < 1$  and  $u(\omega) = 1/2$ . A plot of the sigmoid utility function is shown in Figure 1.

The provider price could be a linear function of the amount of resources:

$$p(r) = \xi \cdot r$$

where  $\xi$  is the *unit price*. The unit price of the resources can be set by convention to a constant, or it can vary based on supply and demand. The variable unit price  $\xi$  might be (a) subject to a peer-to-peer negotiation between the consumer and the provider, (b) set in a centralized way similar to a commodity exchange, requiring global information about the supply and demand, or (c) determined by local estimate of the supply and demand. For example, based on the ratio of the allocated resources to the total resources of the provider, a function could give a lower price for the low ratio and a higher price for the high ratio.

A consumer satisfaction function takes into account both the utility provided to the consumer and the price paid. For a given utility, the satisfaction function should increase when the price decreases and, for a given price, the satisfaction function should increase when the utility  $u$  increases. These requirements are reflected in Equation (3).

$$\frac{\partial s}{\partial p} \leq 0, \quad \frac{\partial s}{\partial u} \geq 0 \quad (3)$$

Furthermore, a normalized satisfaction function should satisfy the following conditions:

- the degree of satisfaction,  $s(u(r), p(r))$ , for a given price  $p(r)$ , approaches the minimum, 0, when the utility,  $u(r)$ , approaches 0;
- the degree of satisfaction,  $s(u(r), p(r))$ , for a given price  $p(r)$ , approaches the maximum, 1, when the utility,  $u(r)$ , approaches infinity;
- the degree of satisfaction,  $s(u(r), p(r))$ , for a given utility  $u(r)$ , approaches the maximum, 1, when the price,  $p(r)$ , approaches 0; and
- the degree of satisfaction,  $s(u(r), p(r))$ , for a given utility  $u(r)$ , approaches the minimum, 0, when the price,  $p(r)$ , approaches infinity.

These requirements are reflected by Equations (4) and (5).

$$\forall p > 0, \lim_{u \rightarrow 0} s(u, p) = 0, \quad \lim_{u \rightarrow \infty} s(u, p) = 1 \quad (4)$$

$$\forall u > 0, \lim_{p \rightarrow 0} s(u, p) = 1, \quad \lim_{p \rightarrow \infty} s(u, p) = 0 \quad (5)$$

A candidate satisfaction function is [19]:

$$s(u, p) = 1 - e^{-\kappa \cdot u^\mu \cdot p^{-\epsilon}} \quad (6)$$

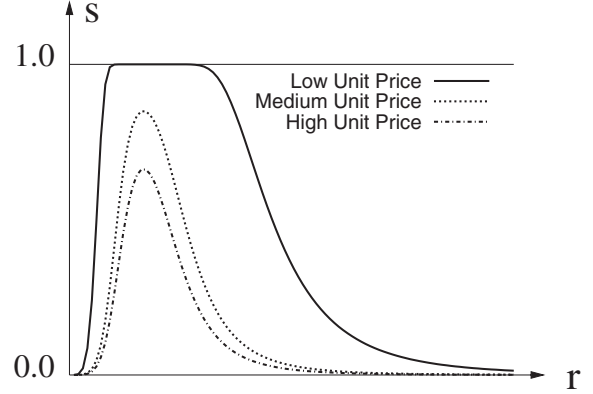
where  $\kappa$ ,  $\mu$ , and  $\epsilon$  are appropriate positive constants.

If in the above satisfaction function the utility is normalized, e.g., the sigmoid specified by Equation 2, it is reasonable to let the price part be also normalized. A candidate satisfaction function that has a normalized price part is

$$s(u, p) = 1 - e^{-\kappa \cdot u^\mu \cdot (p/\phi)^{-\epsilon}} \quad (7)$$

where  $\phi$  is a reference price.

A plot of the satisfaction function based on different unit prices is shown in Figure 2. Satisfaction decreases after a peak value because continuing to pay more as



**Figure 2. The satisfaction function of the amount of resources  $r$  for a sigmoid utility function and linear price functions,  $0 \leq s \leq 1$ . For the same amount of resources, the higher is the price, the lower is the satisfaction.**

resources increase after that point does not increase utilization.

In this paper, we consider a model where the allocation of resources is determined by their price, their utility to the consumer, and by the satisfaction of the consumer. We present the architecture of the model, the utility, price and satisfaction functions, as well as a set of broker strategies. A set of simulations show that the approach can successfully bridge the conflict of interests between the consumers and resource providers and control the global behavior of the system.

## 2 Related Work

Several commercial companies such as Entropia, ProcessTree, Popular Power, Mojo Nation, United Devices, and Parabon are exploiting idle CPU cycles of desktop machines to build computational grids. They are able to charge consumers for access to resources, but do not offering fiscal incentive to all resource contributors. In a long term, it is unlikely that such a model could be successful to create a maintainable and sustainable infrastructure [5].

Economic models are at a better position - resource providers are encouraged because they can benefit from allocating resources to consumers. Several economic models are used for trading computational resources [7]:

- *Commodity Market*: resource providers advertise

their resource prices and charge users based on the amount of resources used. The pricing policy could be based on a flat fee, the resource usage duration, the subscription, and the demand and supply [14]. Mungi [12], Enhanced MOSIX [1], and Nimrod/G [6] are some of the systems based upon the commodity market model.

- *Posted Price*: this model, used by Nimrod/G [6], is similar to the Commodity Market Model except that it advertises special offers to attract users.

- *Bargaining*: resource owners and users/brokers negotiate with each other until they reach a mutually agreeable price. This model is mostly used in a market that does not have a clear demand-and-supply relationship and price. Examples of systems include Mariposa [13] and Nimrod/G [6].

- *Tendering/Contract-Net*: first users/brokers advertise their requirement, then resource owners respond with their bids, and at last users/brokers choose a resource owner to use its resource. Mariposa [13] applies this strategy.

- *Auction*: resource owners announce their resources and invite bids, then an auction process is performed with users/brokers, and at last the winner user/broker uses the resource. Different auction policies can be used: 1) English auction; 2) first-price sealed-bid auction; 3) Vickrey auction (second-price sealed-bid); and 4) Dutch auction. In English auction, all bidders are free to increase their bids exceeding other offers; when no bidder is willing to increase the bid, the auction ends and the highest bidder wins. In first-price sealed-bid auction, every bidder submits a sealed-bid and the highest bidder wins. In Vickrey auction, every bidder submits a sealed-bid and the highest bidder wins at the price of the second highest bidder. In Dutch auction, the resource owners start by a high price and continuously decrease the price until a bidder is will to take the resource at the current price. Spawn [20] and Popcorn [16] use this model.

- *Bid-based Proportional Resource Sharing*: the percentage of resources allocated a user is a function of the user's bid and other users' bids. Rexec/Anemone [8] implements this model.

- *Community/Coalition/Bartering*: a community of resource owners share each other's resources. Those resource owners contribute to the community get credits by sharing their resources. The credit of a resource owner decides how much resources he can get from others. Condor [9], SETI@home [17], and Mojo Nation [15] are based on this model.

- *Monopoly/Oligopoly*: one or a small number of resource owners decide the price and it is not possible to negotiate the price. Nimrod/G [6] embraces this model.

Some systems use more than one strategy. For example, Nimrod/G supports three different models: the Commodity Market model, the Posted Price model, and the Monopoly/Oligopoly model.

The efficiency of two different market-based resource allocation schemes, commodities markets and auctions, are discussed in [21] and [22]. These papers define concepts such as price stability, market equilibrium, consumer efficiency, and provider efficiency and show that the commodities markets are better choices for controlling grid resources than auction strategies.

An approach to implement automatic selection of multiple negotiation models to adapt to the computation needs and change of resource environment is discussed in [18]. A task-oriented mechanism for measuring the economic value of using heterogeneous resources as a common currency is analyzed in [11]; resource consumers can compare the advantage of participating in a computational grid with the alternative of purchasing their own resources necessary, and resource providers can evaluate the profit of putting their resources into a grid. A comparative analysis of market-based resource allocation by continuous double auctions and by the proportional share protocol versus a conventional round-robin approach is presented in [10].

### 3 Broker Models

The role of a broker is to facilitate the resource allocation. We do not support task migration. Once allocated to a site, we expect the task either to complete its execution there, or if it fails to complete at that site, to be restarted at a different site. We also assume that a matchmaking service has previously been invoked to establish that indeed the resources required by a consumer  $U_i$  exist at site  $R_j$ .

Several broker models can be constructed subject to these restrictions. For example, we could consider *specialized brokers* that mediate access to sets of "similar" resources, e.g., one broker is responsible for all clusters, another for high performance parallel systems, one for systems with graphical capabilities, and so on. An alternative model where individual brokers mediate access to any type of resources is possible. We could also construct models in which one consumer is allowed to request services from one broker only; alternatively one could construct models where one consumer requests services from competing brokers.

The *provider-broker-consumer model* involves the set of resource providers  $\mathcal{R}$ , the set of consumers  $\mathcal{U}$ , and broker  $B$ . Brokers have “societal goals” and attempt to maximize the average utility and revenue, as opposed to providers and consumers that have individualistic goals; each provider wishes to maximize its revenue, while each consumer wishes to maximize its utility and do so for as little cost as possible. To reconcile the requirements of a consumer and the candidate providers, a broker first chooses a subset of providers such that the satisfaction is above a high water mark and all providers in the subset have equal chances to be chosen by the consumer. We call the size of this subset *satisficing size*, and denote it as  $\sigma$ . The model consists of the following steps (Figure 3):

1. All resource providers reveal their capacity and pricing parameters to the broker:  $\forall R_j \in \mathcal{R}$  send  $c_j$  and  $\xi_j$ , where  $c_j$  and  $\xi_j$  are the resource capacity and the unit price of  $R_j$ , respectively.
2. A consumer  $U_i$  sends to the broker: (1) the parameters of its utility function:  $(\zeta_i, \omega_i)$ , (2) the parameters of its satisfaction function:  $(\mu_i, \epsilon_i, \kappa_i, \phi_i)$ , and (3) the maximum number of candidate resource providers to be returned.
3. The broker performs an algorithm and returns a list of candidate resource providers  $\mathcal{R}^i$  to consumer  $U_i$ .
4. Consumer  $U_i$  selects the first provider from  $\mathcal{R}^i$  and verifies if the provider can allocate the required resources. If it can not, the consumer moves to the next provider from the list until the resources are allocated by a provider  $R_j$ .
5.  $R_j$  notifies the broker about the resource allocation to  $U_i$ .

The brokering algorithm is summarized in Figure 4. The amount of resources to be allocated is determined during the algorithm according to a *broker strategy*. Simple strategies would be to allocate the same amount of resources to every consumer, or to allocate to every consumer a random amount of resources. A better strategy, discussed in this paper, is to allocate an amount of resources such that the utility of the resource to the consumer reaches a certain *target utility*  $\tau$ . To determine the amount of resources allocated to the consumer, the broker uses Equation 8 derived from the definition of  $u(r)$ , Equation 2:

$$r = e^{\frac{\ln(\frac{\tau}{1-\tau})}{\zeta} + \ln(\omega)} \quad (8)$$

Several quantities are used in the next section to characterize the resource management policy for broker  $B$  and its associated providers and consumers:

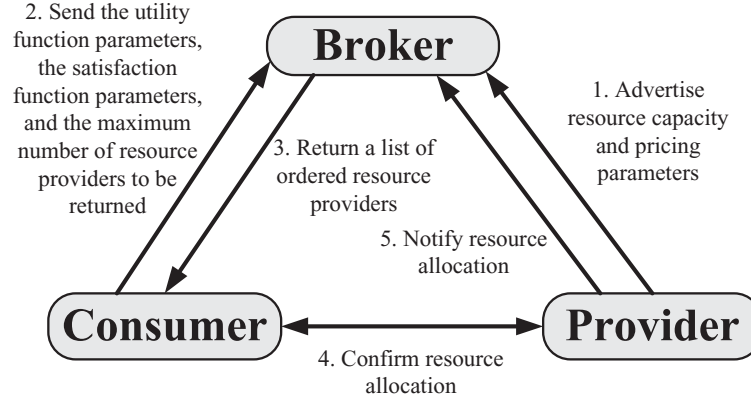
1. The average hourly revenue for providers. The revenue of a provider is over all of its resources. This average is over the set of all providers connected to  $B$ .
2. The consumer admission ratio. This ratio is the number of admitted consumers over the number of all consumers connected to  $B$ . A consumer is admitted into the system when there is a provider able to allocate resources, otherwise the consumer is dropped.
3. The average consumer utility. This average is over the set of all admitted consumers connected to  $B$ .
4. The average consumer satisfaction. This average is over the set of all admitted consumers connected to  $B$ .

## 4 A Simulation Study

Market oriented resource allocation algorithms are very difficult to analyze analytically. To understand the transient and the steady state behavior of the system we conducted a simulation study using YAES [24, 4]. The behavior of the model is determined by two parameters that can be chosen by the broker: the target utility  $\tau$  and the satisficing size  $\sigma$ . We investigate the performance of the model for different  $\tau$  and  $\sigma$  values. When we study the effect of  $\tau$ , we use  $\sigma = 1$ , and when we study the effect of  $\sigma$ , we use  $\tau = 0.9$ . We also compare the system performance of our scheme for several  $\sigma$  values with a *random strategy*. In this case, we randomly choose a provider from the set of all providers, without considering the satisfaction function.

We define the *demand-capacity ratio* as the ratio of the amount of resources requested by the consumers to the total capacity of resource providers. While the capacity can be exactly determined, the level of demand is practically limited by the sigmoid shape of the utility curve. In the computation of the demand-capacity ratio, for each consumer and each resource, it is assumed that for the requested resource the corresponding utility value is 0.9. We investigate the performance of the model in the steady-state under various scenarios of demand-capacity ratio for different  $\tau$  and  $\sigma$  values.

In our simulation, the system consists of 100 providers and a broker. The capacity of providers is a random variable normally distributed with the mean of



**Figure 3. The brokering process: 1) Providers send to the broker their resource capacity and pricing parameters; 2) A consumer sends to the broker a request; 3) The broker executes a brokering algorithm and returns to the consumer a list of resource providers; 4) The consumer selects the first provider from the returned list and confirms that it can satisfy the resource requirements; if it can not, chooses the next provider, until one of the providers allocates the needed resources; 5) If a resource provider allocates resources to the consumer, it notifies the broker about this allocation.**

#### BROKERING ALGORITHM

INPUT request  $req$ ,  $\tau$ ,  $\sigma$ , a finite set of resource providers  $ps$

OUTPUT a finite set of suggested resource providers  $ss$

BEGIN

determine  $amount$  so that  $req.u(amount) = \tau$

FOR each resource provider  $rp$  in  $ps$

$r = \min(amount, \text{available resources of } rp)$

$satisfaction = req.s(req.u(r), rp.p(r))$

END FOR

sort elements in  $ps$  according to their  $satisfactions$

randomize the sequences of the first  $\sigma$  items in  $ps$

keep the elements in  $ps$  that have the highest  $req.cardinality$  satisfaction degrees and remove the rest

$ss = ps$

END

**Figure 4. The brokering algorithm performed by the broker.  $req$  contains a utility function  $u$ , a satisfaction function  $s$ , and a  $cardinality$  that specifies the number of resource providers to be returned by the broker.  $\tau$  is the target utility.  $\sigma$  is the satisficing size.**

1000 and the standard deviation of 500. All providers use a linear pricing strategy (see Table 1). Initially, there is no consumer in the system. The inter-arrival time of consumers follows an exponential distribution

with the mean of 2 seconds. The service time, i.e., the time a consumer uses the resource allocated to it, follows an exponential distribution with the mean of  $\lambda$  seconds. By varying the  $\lambda$  parameter we modify

Parameter	Range
$\xi$	[1, 10]
$\omega$	[20, 40]
$\kappa$	[0.02, 0.04]
$\mu$	[2, 4]
$\epsilon$	[2, 4]
$\phi$	[900, 1000]

**Table 1. Parameters of the simulation with a uniform distribution.**

demand-capacity ratio so that we can study the behavior of the system under different loads. The parameters of the utility function of consumers are uniformly distributed in the ranges shown in Table 1. With these values, for the fixed value of utility of 0.9, the resource demands of the consumers are exponentially distributed with mean of 500. For simplification we let the maximum number of resource providers to be returned by the broker be 100 for all consumers.

In every case, we run the simulation 50 times. We show the average value and a 95% confidence interval. We report the average hourly revenue, the consumer admission ratio, the average consumer satisfaction, and the average consumer utility collected over the most recent one hour interval:

1. As a function of time: for several levels of target utility  $\tau$  (Figure 5, 6, 7, and 8); for several levels of satisficing size  $\sigma$  (Figure 9, 10, and 11).
2. As a function of the demand-capacity ratio: for several levels of target utility  $\tau$  (Figure 12, 13, 14, and 15); for several levels of satisficing size  $\sigma$  (Figure 16, 17, 18, and 19).

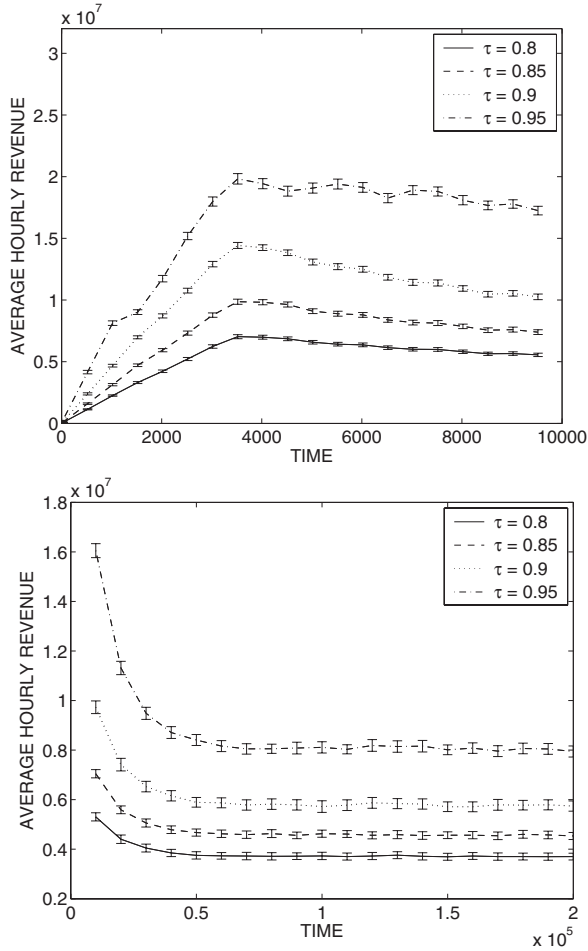
Figure 5 shows that the average hourly revenue increases rapidly during the transient period, followed by a slow decrease due to resource fragmentation<sup>1</sup>, and reaches a steady value for the steady state. The larger is  $\tau$ , the more resources are allocated to consumers, and the higher is the average hourly revenue. Figure 6 shows that when  $\tau$  is set to 0.8, 0.85, or 0.9, the system is capable of handling all consumer requests. So for these  $\tau$  values the consumer admission ratio is approximately 1.0 and the corresponding plots overlap with each other. When  $\tau$  is equal to 0.95, the system is not capable of handling all consumer requests in

<sup>1</sup>Resource fragmentation is an undesirable phenomena; in our environment the amount of resources available from any single provider cannot meet the target utility value for any request and an insufficient amount of resources is allocated.

the transient period, and some consumer requests are dropped. As time goes on, due to resource fragmentation, many consumers are allocated smaller amounts of resources (below the one required by the target utility). Thus, more consumers can be admitted to the system, increasing the consumer admission ratio. However, these additional consumers would operate at lower levels of utility and satisfaction. During the steady state, all consumers can be admitted and the consumer admission ratio is 1. Figure 7 shows that the average consumer satisfaction increases slowly during the transient period and reaches a stable value in steady state. The average consumer satisfaction is higher when  $\tau$  is smaller because the smaller is  $\tau$ , the more consumers can be admitted by resource providers with cheaper prices and these consumers experience higher satisfactions. Figure 8 shows that at the startup of the system, with all the providers available, the early customers will have a relatively higher average utility (although limited by the value of  $\tau$ ). The average utility slowly decreases to a stable value during the steady state, which shows the adaptation of the system to the current load conditions. The average consumer utility is lower when  $\tau$  is smaller.

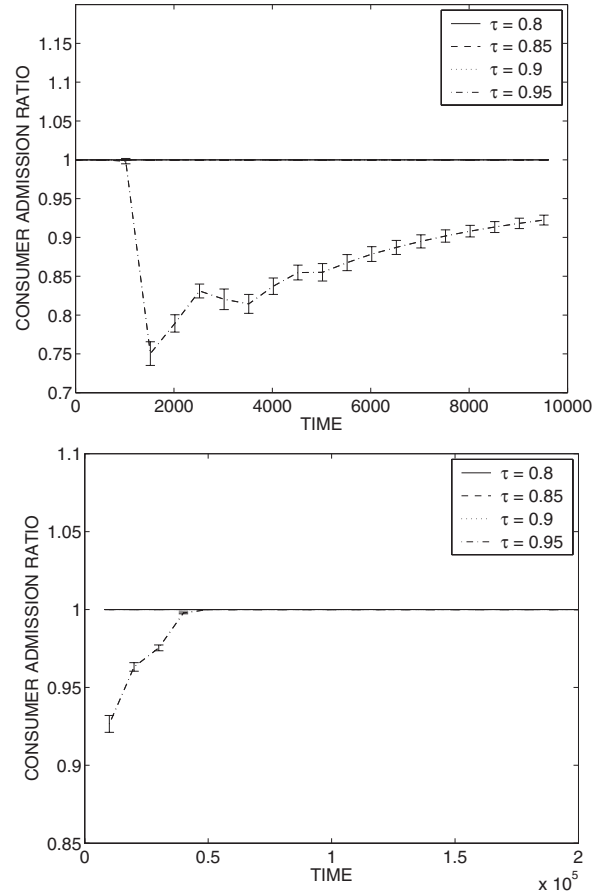
Figure 9 shows that the average hourly revenue increases rapidly during the transient period. It decreases slowly due to resource fragmentation after the transient period and leads to a stable value in steady state. A small value of  $\sigma$  limits the number of choices the broker has and this restriction leads to lower average hourly revenues. The larger is  $\sigma$ , the higher is the average hourly provider revenue. The random strategy, which corresponds to the maximum value of  $\sigma = |\mathcal{R}|$ , has the highest average hourly provider revenue. Figure 10 shows that after the unpredictable early stages of the transient period, the average consumer satisfaction increases gradually to a stable value in steady state. The average consumer satisfaction is higher when  $\sigma$  is smaller. Indeed, when  $\sigma = 1$  we direct the consumer to that resource provider that best matches the request. When we select at random one provider from the set of all providers we observe the lowest average consumer satisfaction. Indeed, when we resort to a random strategy we have a high probability to select a less than optimal match for a given request. The optimal match is the top ranked element of the candidate resource provider list. Figure 11 shows that the average consumer utility is lower when  $\sigma$  is smaller; the random strategy has the highest average consumer utility because it has the largest  $\sigma$ . When  $\sigma$  is larger consumers have a better chance to get resources according to the  $\tau$  values.

Figure 12 shows that as the demand-capacity ratio



**Figure 5. Average hourly revenue vs. time (in seconds) for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . Top: transient period. Bottom: steady state. The larger is  $\tau$ , the more resources are allocated to consumers, and the higher is the average hourly revenue.**

increases, the system experiences a transition from a lightly loaded to a heavily loaded state, and eventually reaches saturation. The average hourly revenue increases and reaches a steady value when the system is saturated. The stable values for different  $\tau$  values are the same. The larger is the  $\tau$ , the earlier the average hourly revenue reaches the stable value because the more resources are allocated to consumers and the earlier the system reaches saturation. Figure 13 shows that the consumer admission ratio for a lightly loaded system is 1; as the load increases some consumers are

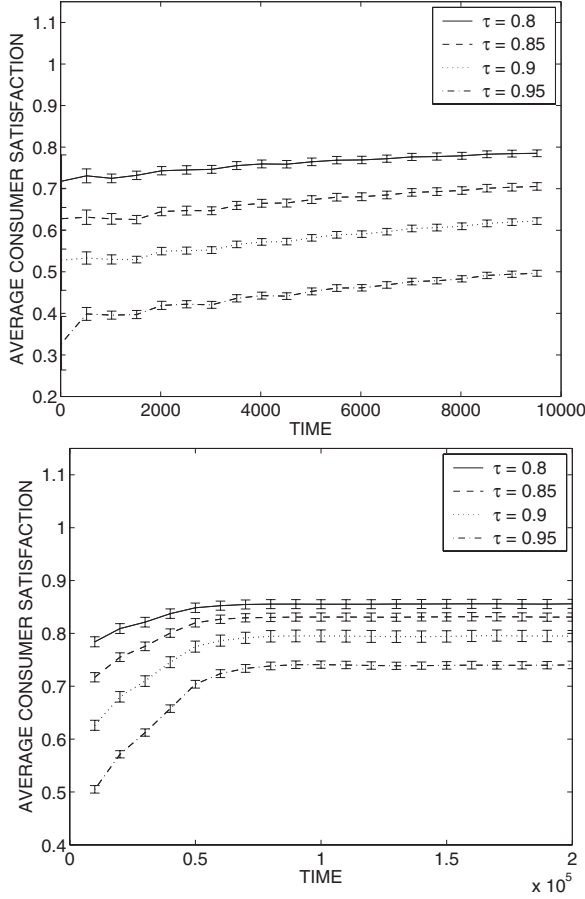


**Figure 6. Consumer admission ratio vs. time (in seconds) for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . Top: transient period. Bottom: steady state.**

rejected and the ratio drops. The larger is  $\tau$ , the earlier the system starts rejecting consumers and the admission ratio starts the drop. Figure 14 shows that for a lightly loaded system the average consumer satisfaction increases when the demand-capacity ratio increases until it reaches a steady value. With a smaller  $\tau$  the consumers are allocated smaller, cheaper blocks of resources. As long as the resource allocation is sufficiently large, the cheaper allocation leads to a higher satisfaction. Figure 15 shows that for a lightly loaded system the average consumer utility decreases until it reaches a steady value as the demand-capacity ratio increases. The smaller is  $\tau$ , the lower is this steady value.

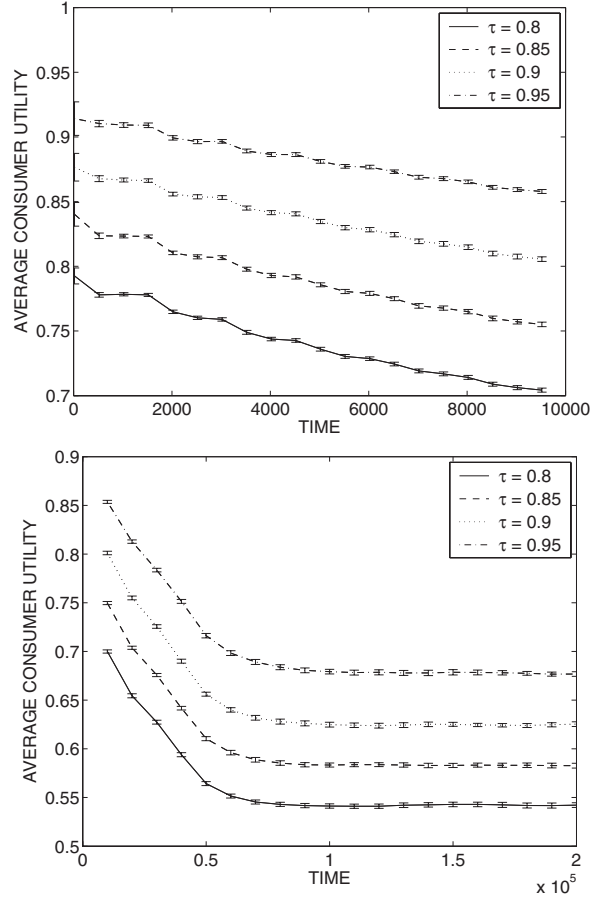
Figure 16 shows that the stable values for different  $\sigma$





**Figure 7. Average consumer satisfaction vs. time (in seconds) for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . Top: transient period. Bottom: steady state. The average consumer satisfaction is higher when  $\tau$  is smaller.**

values are the same. The larger is  $\sigma$ , the earlier the stable value is reached. The random strategy is the first one to reach the steady value. Figure 17 shows that the smaller is  $\sigma$ , the earlier the system starts rejecting consumers. Figure 18 shows that before saturation, the smaller is  $\sigma$ , the higher is the average consumer satisfaction because consumers have more chance to get resources from providers offering cheaper prices (recall that average consumer satisfaction is over only admitted consumers). After saturation, the larger is  $\sigma$ , the higher is the average consumer satisfaction. The random strategy has the largest steady consumer satisfaction. Figure 19 shows that before saturation, the smaller is  $\sigma$ , the lower is the average consumer utility

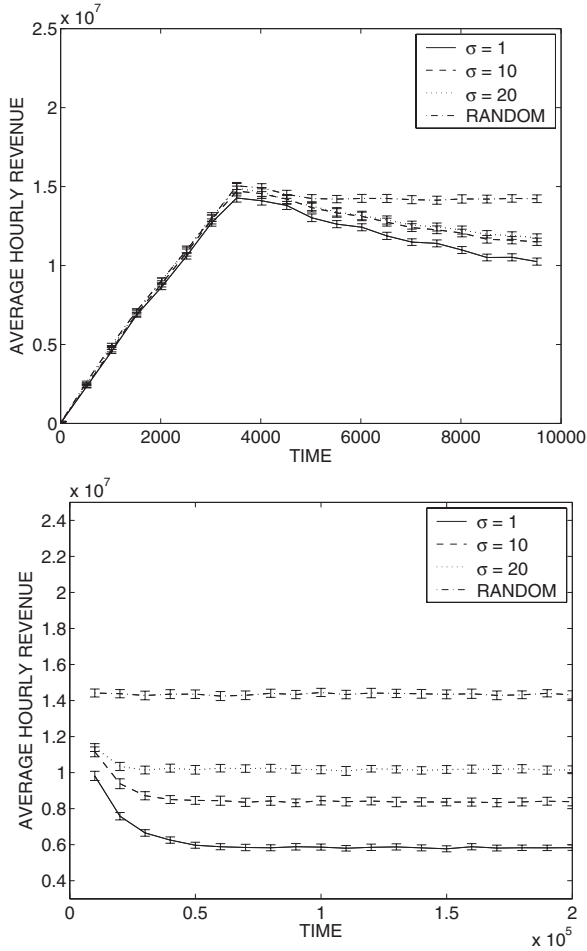


**Figure 8. Average consumer utility vs. time (in seconds) for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . Top: transient period. Bottom: steady state. The average consumer utility is lower when  $\tau$  is smaller.**

because consumers have more chance to be admitted by providers offering cheaper resources with lower utilities. After saturation, the larger is  $\sigma$ , the lower is the average consumer utility because consumers have more chance to be admitted by providers with lower utilities (recall that average consumer utility is based only on admitted consumers).

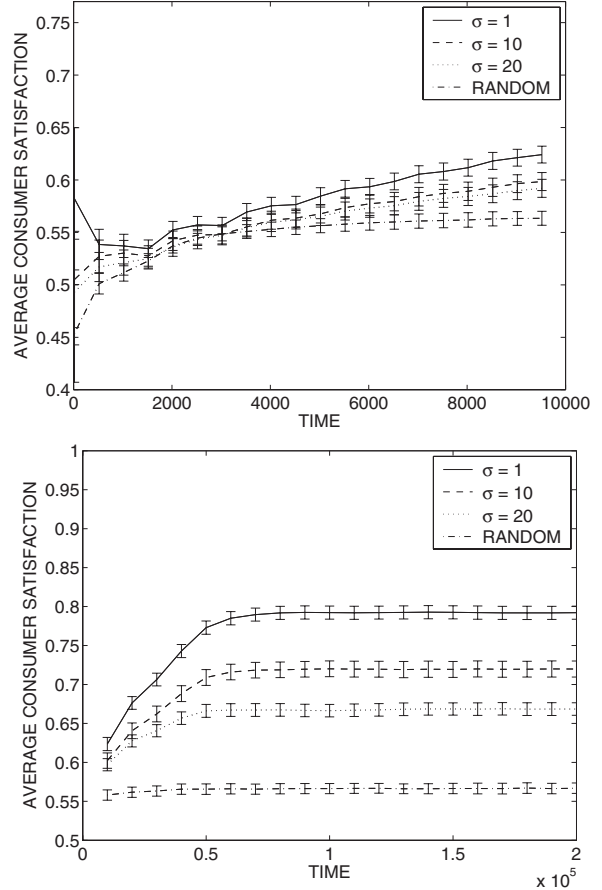
## 5 Conclusions and Future Work

In this paper we discuss the role of a broker in market oriented resource allocation models for large-scale heterogeneous systems. The scale of the system, the heterogeneity of resources, the diversity of the pricing



**Figure 9. Average hourly revenue vs. time (in seconds) for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . Top: transient period. Bottom: steady state. The larger is  $\sigma$ , the higher is the average hourly provider revenue.**

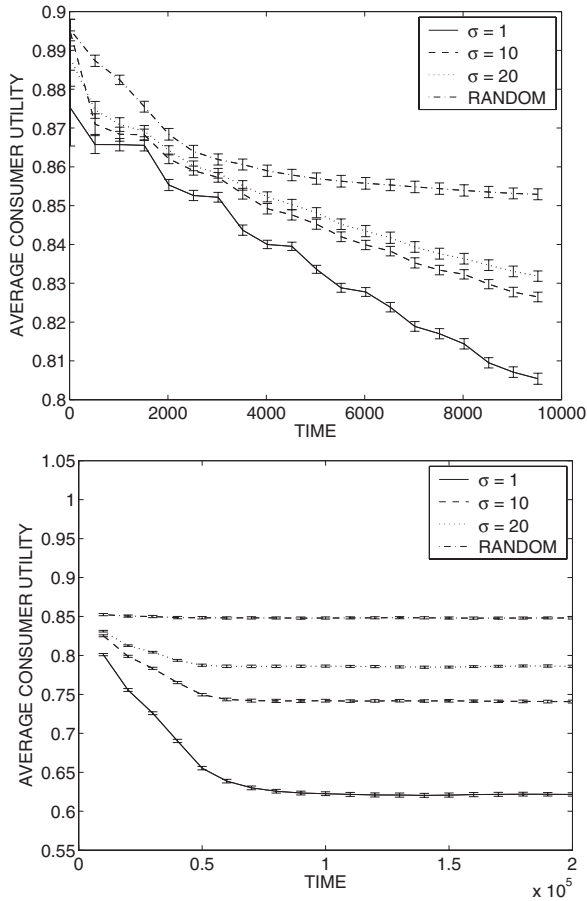
structures, the large number of consumers and the variety of their needs, make such models difficult to analyze. Utility-based resource allocation has been successfully adopted in the area where only one type of resource is used and the user population is limited, such as in [2]. In this paper we are considering the case when either there is only one critical resource in the system or the multiple critical resources can be represented with a synthetic “superresource.” This simplification allows us to derive rules of thumb for the adjustment of broker parameters to improve the performance of the system.



**Figure 10. Average consumer satisfaction vs. time (in seconds) for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . Top: transient period. Bottom: steady state. The average consumer satisfaction is higher when  $\sigma$  is smaller.**

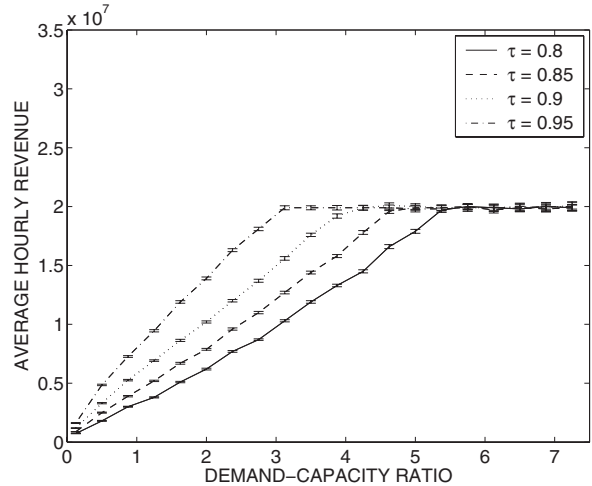
A more complex model based on resource vectors and multi-dimensional utility, price, and satisfaction functions is considered in [3]. In a multi-dimensional resource allocation, however, the resource providers can be only arranged in a partial order, which requires different assumptions.

Our simulation results show that the  $\tau$  and  $\sigma$  parameters can be used to control the global behavior of the system. A broker could monitor the system performance and adjust  $\tau$  and  $\sigma$  values for performance improvement. We are able to anticipate the results of different actions of such a broker. For example, if the broker perceives that the system is overloaded



**Figure 11. Average consumer utility vs. time (in seconds) for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . Top: transient period. Bottom: steady state. The average consumer utility is lower when  $\sigma$  is smaller.**

and many consumers have to be dropped, it has two choices: decrease  $\tau$  or increase  $\sigma$ . If it decreases  $\tau$ , more consumers are admitted, while the system experiences a decrease of average hourly avenue, a decrease of average consumer utility, and an increase of average consumer satisfaction. If the broker increases  $\sigma$  to admit more consumers, there are an increase of average hourly avenue, an increase of average consumer utility, and a decrease of average consumer satisfaction. It seems counterintuitive that when we increase the utility we get lower satisfaction, but it reflects the fact that when we increase the utility the cost may be increased to a level beyond the optimum satisfaction.



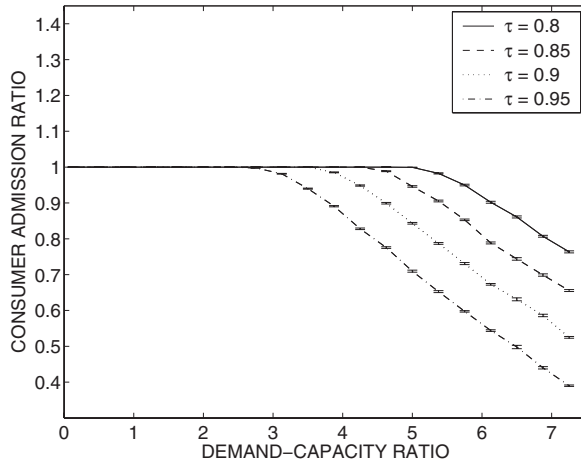
**Figure 12. Average hourly revenue vs. demand-capacity ratio in steady state for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . The larger is the  $\tau$ , the earlier the average hourly revenue reaches the stable value because the more resources are allocated to consumers and the earlier the system reaches saturation.**

## 6 Acknowledgments

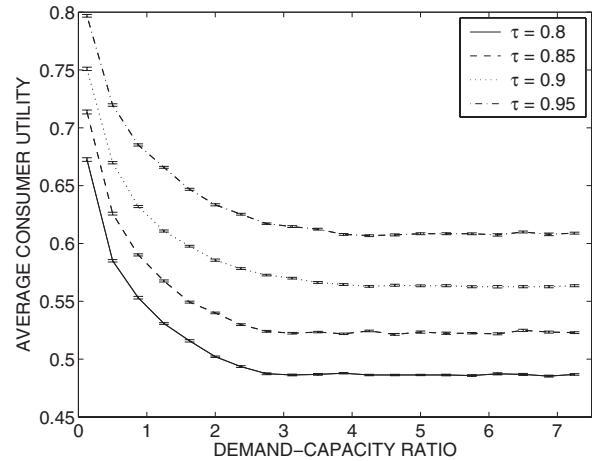
This research was supported in part by National Science Foundation grants MCB9527131, DBI0296035, ACI0296035, and EIA0296179, the Colorado State University George T. Abell Endowment, and the DARPA Information Exploitation Office under contract No. NBCHC030137. The authors thank E. K. P. Chong and A. A. Maciejewski for their comments. Approved for public release, distribution unlimited.

## References

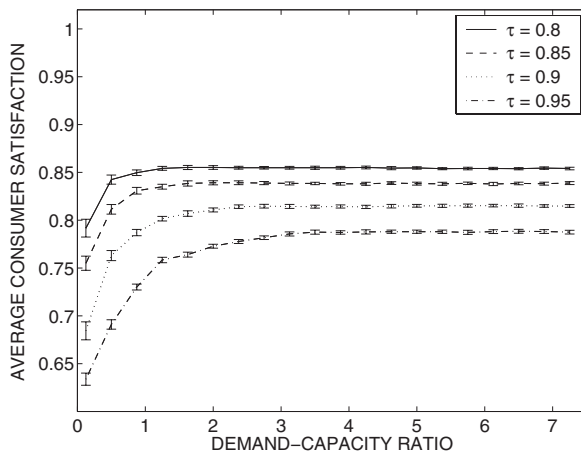
- [1] Y. Amir, B. Awerbuch, A. Barak, R. S. Borgstrom, and A. Keren. An opportunity cost approach for job assignment in a scalable computing cluster. *IEEE Transactions on Parallel and Distributed Systems*, 11(7):760–768, 2000.
- [2] L. Badia and M. Zorzi. On utility-based radio resource management with and without service guarantees. In *Proc. ACM MSWiM 2004, Modelling, Analysis, and Simulation of Wireless and Mobile Systems*, pages 244–251. ACM Press, 2004.
- [3] X. Bai, L. Bölöni, D. C. Marinescu, and H. J. Siegel. Are utility, price, and satisfaction resource allocation



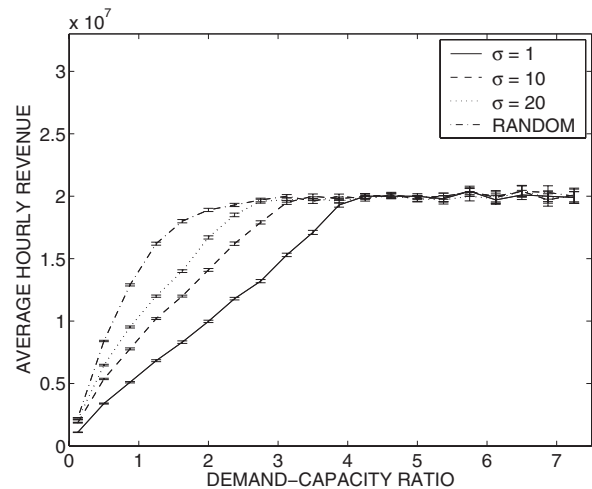
**Figure 13.** Consumer admission ratio vs. demand-capacity ratio in steady-state for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . The larger is  $\tau$ , the earlier the system starts rejecting consumers and the admission ratio starts the drop.



**Figure 15.** Average consumer utility vs. demand-capacity ratio in steady-state for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . The smaller is  $\tau$ , the lower is the steady value.



**Figure 14.** Average consumer satisfaction vs. demand-capacity ratio in steady state for  $\sigma = 1$  and  $\tau = 0.8, 0.85, 0.9,$  and  $0.95$ . The smaller is  $\tau$ , the higher is the steady value.



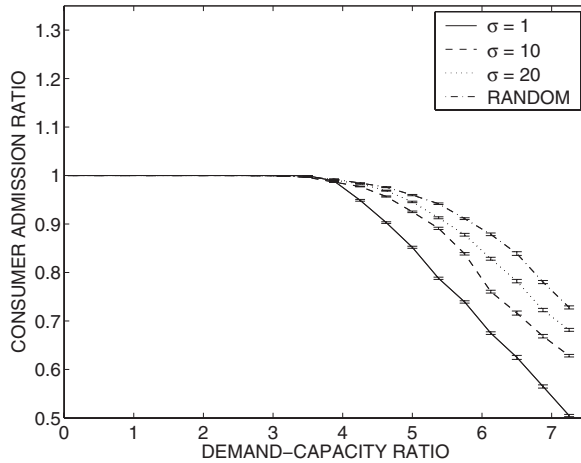
**Figure 16.** Average hourly revenue vs. demand-capacity ratio in steady-state for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . The larger is  $\sigma$ , the earlier the stable value is reached.

models suitable for large-scale distributed systems? In preparation.

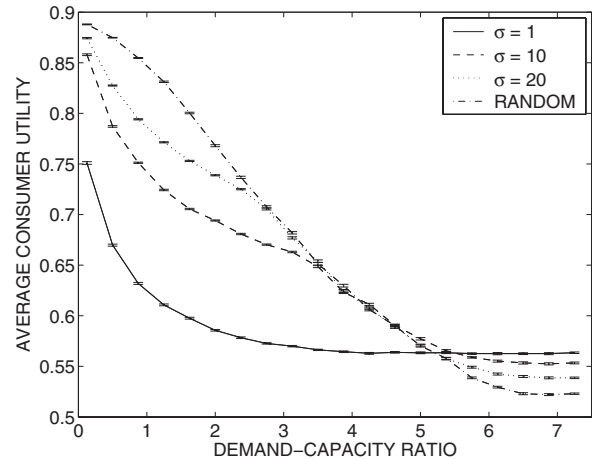
- [4] L. Bölöni and D. Turgut. YAES - a modular simulator for mobile networks. In *Proceedings of the 8-th ACM/IEEE International Symposium on Modeling,*

*Analysis and Simulation of Wireless and Mobile Systems MSWIM 2005*, pages 169–173, October 2005.

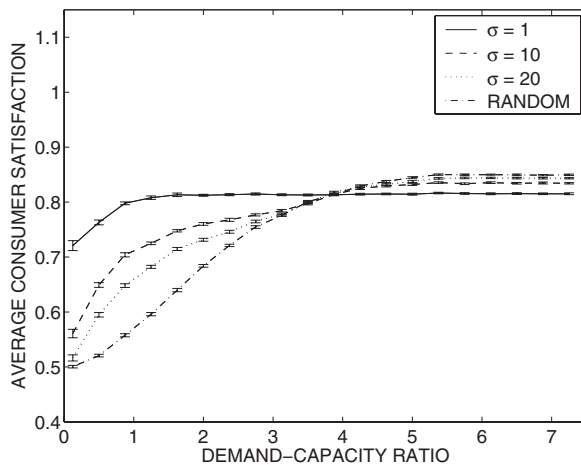
- [5] R. Buyya, D. Abramson, and J. Giddy. A case for economy grid architecture for service-oriented grid computing. In *Proc. of the 15th Int. Parallel & Dis-*



**Figure 17. Consumer admission ratio vs. demand-capacity ratio in steady-state for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . The smaller is  $\sigma$ , the earlier the system starts rejecting consumers.**



**Figure 19. Average consumer utility vs. demand-capacity ratio in steady-state for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . Before saturation, the smaller is  $\sigma$ , the lower is the average consumer utility; after saturation, the larger is  $\sigma$ , the lower is the average consumer utility.**



**Figure 18. Average consumer satisfaction vs. demand-capacity ratio in steady-state for  $\tau = 0.9$  and  $\sigma = 1, 10,$  and  $20$ . For the random strategy,  $\sigma = |\mathcal{R}| = 50$ . Before saturation, the smaller is  $\sigma$ , the higher is the average consumer satisfaction; after saturation, the larger is  $\sigma$ , the higher is the average consumer satisfaction.**

*tributed Processing Symposium (IPDPS-01)*, page 83, 2001.

- [6] R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: An architecture of a resource management and scheduling system in a global computational grid. In *Proc. of the 4th Int. Conf. on High Performance Computing in the AsiaPacific Region*, volume 1, pages 283–289, 2001.
- [7] R. Buyya, H. Stockinger, J. Giddy, and D. Abramson. Economic models for management of resources in peer-to-peer and grid computing. In *Proc. of the SPIE Int. Conf. on Commercial Applications for High-Performance Computing*, pages 13–25, Denver, USA, August 20–24 2001.
- [8] B. Chun and D. Culler. Market-based proportional resource sharing for clusters. Technical report, University of California, Berkeley, September 1999.
- [9] CONDOR. URL <http://www.cs.wisc.edu/condor/>.
- [10] J. Gomoluch and M. Schroeder. Market-based resource allocation for grid computing: A model and simulation. In *Int. Middleware Conf., Workshop Proc.*, pages 211–218, Rio de Janeiro, Brazil, June 16–20 2003.
- [11] L. He and T. R. Ioerger. Task-oriented computational economic-based distributed resource allocation mechanisms for computational grids. In *Proc. of the Int. Conf. on Artificial Intelligence*, volume 1, pages 462–468, Las Vegas, Nevada, USA, June 21–24 2004.
- [12] G. Heiser, F. Lam, and S. Russel. Resource management in the mungi single-address-space operating sys-

- tem. In *Proc. of the 21 st Australasian Computer Science Conf.*, pages 417–428, February 1998.
- [13] Mariposa. URL <http://mariposa.cs.berkeley.edu/>.
- [14] L. W. McKnight and J. Boroumand. Pricing internet services: Approaches and challenges. *IEEE Computer*, 33(2):128–129, 2000.
- [15] Mojo Nation. URL <http://www.mojonation.net/>.
- [16] N. Nisan, S. London, O. Regev, and N. Camiel. Globally distributed computation over the internet - the popcorn project. In *ICDCS '98: Proc. of The 18th Int. Conf. on Distributed Computing Systems*, pages 592–601, Washington, DC, USA, 1998. IEEE Computer Society.
- [17] SETI@home. URL <http://setiathome.ssl.berkeley.edu/>.
- [18] W. Shen, Y. Li, H. H. Genniwa, and C. Wang. Adaptive negotiation for agent-based grid computing. *Journal of the American Statistical Association*, 97(457), 2002.
- [19] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. Norton, New York, March 1999.
- [20] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *Software Engineering*, 18(2):103–117, 1992.
- [21] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. Analyzing market-based resource allocation strategies for the computational Grid. *The Int. Journal of High Performance Computing Applications*, 15(3):258–281, Fall 2001.
- [22] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. G-commerce: Market formulations controlling resource allocation on the computational grid. In *Proc. of the 15th Int. Parallel & Distributed Processing Symposium (IPDPS-01)*, pages 23–27, San Francisco, CA, April 2001.
- [23] M. Xiao, N. Shroff, and E.-P. Chong. Utility-based power control in cellular wireless systems. In *INFOCOM 2001, Joint Conf. of the IEEE Computer and Communication Societies*, pages 412–421, 2001.
- [24] YAES. URL <http://netmoc.cpe.ucf.edu/>.

## Biographies

**Xin Bai** is a Ph.D. candidate in the School of Electrical Engineering and Computer Science at the University of Central Florida (UCF). He received a B.S. degree from Northern Jiaotong University in 1993 and an M.S. degree from the University of Central Florida in 2003. His research areas include grid computing and large-scale distributed systems.

**Ladislau Bölöni** is an assistant professor at the Computer Engineering department of University of Central Florida. He received a PhD degree from the Computer Sciences Department of Purdue University

in May 2000. He received a Master of Science degree from the Computer Sciences department of Purdue University in 1999 and Diploma Engineer degree in Computer Engineering with Honors from the Technical University of Cluj-Napoca, Romania in 1993. He received a fellowship from the Hungarian Academy of Sciences for the 1994-95 academic year. He is a member of ACM and the Upsilon Pi Epsilon honorary society. His research interests include distributed object systems, autonomous agents and parallel computing.

**Dan C. Marinescu** is Professor of Computer Science at University of Central Florida. He is a Provost Research Professor and Scientific Director of the Interdisciplinary Information Science and Technology Laboratory at UCF. From 1984 till 2001 he was a Professor in the Department of Computer Science at Purdue University. He is conducting research in parallel and distributed computing, computational structural biology, and quantum computing. He has co-authored more than 160 papers published in refereed journals and conference proceedings. He is the author of the book *Internet-based Workflow Management* published by Wiley in 2002, has co-edited *Process Coordination and Ubiquitous Computing* published by CRC Press in 2002. His most recent book *Approaching Quantum Computing* was published by Prentice Hall in 2004. He is a member of the editorial board of several journals and member of the Program Committee for international conferences. He was the keynote speaker and tutorial lecturer at scientific meetings in US and abroad. He has consulted for industry and government. See <http://www.cs.ucf.edu/~dcm> for additional information.

**Howard Jay Siegel** holds the endowed chair position of Abell Distinguished Professor of Electrical and Computer Engineering at Colorado State University (CSU), where he is also a Professor of Computer Science. He is the Director of the CSU Information Science and Technology Center (ISTeC). ISTeC a university-wide organization for promoting, facilitating, and enhancing CSU's research, education, and outreach activities pertaining to the design and innovative application of computer, communication, and information systems. Prof. Siegel is a Fellow of the IEEE and a Fellow of the ACM. From 1976 to 2001, he was a professor in the School of Electrical and Computer Engineering at Purdue University. He received a B.S. degree in electrical engineering and a B.S. degree in management from the Massachusetts Institute of Technology (MIT), and the M.A., M.S.E., and Ph.D. degrees from the Department of Electrical Engineering and Computer Science at Princeton University. He has co-authored over 300 technical papers. His research in-

terests include heterogeneous parallel and distributed computing, parallel algorithms, parallel machine interconnection networks, and reconfigurable parallel computer systems. He was a Coeditor-in-Chief of the Journal of Parallel and Distributed Computing, and has been on the Editorial Boards of both the IEEE Transactions on Parallel and Distributed Systems and the IEEE Transactions on Computers. He was Program Chair/Co-Chair of three major international conferences, General Chair/Co-Chair of six international conferences, and Chair/Co-Chair of five workshops. He is a member of the Eta Kappa Nu electrical engineering honor society, the Sigma Xi science honor society, and the Upsilon Pi Epsilon computing sciences honor society. He has been an international keynote speaker and tutorial lecturer, and has consulted for industry and government. For more information, please see [www.engr.colostate.edu/~hj](http://www.engr.colostate.edu/~hj).

**Rose Daley** is a member of the Senior Professional Staff at the Johns Hopkins University Applied Physics Lab (JHU/APL). She holds a B.S. in Electrical Engineering from Rensselaer Polytechnic Institute and an M.S. in Computer Science from the John Hopkins University, specializing in Distributing Computing. She has over twenty years experience architecting and implementing software systems, including both distributed large-scale tactical systems encompassing multiple operating systems and communication protocols, and enterprise systems with large databases on internal Intranets. She is the PI for a project for adaptive middleware in large-scale computing environments as well as two internal efforts for developing specialized architecture frameworks. She has led numerous efforts in architecture and modeling, resource management, determining tactical mission sensitivity to network resource attacks and casualties, including development of several tactical and enterprise DoD systems.

**I-Jeng Wang** is a Principal Professional Staff with the Research and Technology Development Center at the Johns Hopkins University Applied Physics Laboratory. He has a joint appointment with the Johns Hopkins University Computer Science Department as a Research Assistant Professor. He received the M.S. degree from Penn State University in 1991 and the Ph.D. degree from Purdue University in 1996, both in Electrical Engineering. From 1996 to 1997, he was a postdoctoral fellow with the Institute for Systems Research at the University of Maryland, where he conducted research in intelligent control and stochastic approximation. Since October 1997, he has been with JHU/APL where he manages and directs internal research in developing scalable algorithms for solving large-scale DoD problems in areas including resource allocation, wireless networking and pattern recognition. He is the Co-PI of a project on mission-oriented resource management for a Total Ship Computing Environment funded by the DARPA IXO ARMS program. He was the PI of a project on adaptive information control to develop efficient resource allocation techniques for dynamic QoS provisioning over distributed and disparate networks, funded by the DARPA AICE program. He was a Co-PI of a project on autonomous internetworking funded by the Army Collaborative Technology Alliance, under which he leads a multi-organization team to develop scalable and dynamic routing protocols. His current research interests include stochastic optimization and control, sensor networks, wireless networking, and distributed Bayesian learning and inference. He is an associate editor for the *IEEE Transactions on Automatic Control*.