Experiments in Designing Computational Economies for Mobile Users

Tracy Mullen [a] Jack Breese [b]

[a] *Artificial Intelligence Laboratory*
University of Michigan
Ann Arbor, MI 48109-2110 USA

[b] *Microsoft Research*
Redmond WA 98052-6399 USA

Abstract

Distributed operating systems provide users with transparent access to network-wide resources. As changes occur to network resources or as user locations and preferences evolve, the system must be able to adapt by reallocating, replicating, or moving its resources. We explore a market-based approach to resource allocation in such environments, and describe initial experiments in designing and building computational markets for distributed operating system resources. We focus on the problem of redistributing network file allocations for a mobile user population in order to improve locality and accessibility.

Key words: Computational economies; distributed resource allocation; market-based design; multi-agent systems.

1 Introduction

As computing becomes more ubiquitous, distributed operating systems must deal with a user population that is increasingly distributed and mobile. Hiding the complexity of such a computing environment from the user requires the operating system to support more location transparent, adaptable and self-tuning services [3].

In this paper, we examine a market-based approach to allocating network-wide files to a mobile user population based on locality and accessibility. Since a market price system is essentially solving a distributed optimization problem for multiple resource allocation, it provides a natural framework for addressing resource allocation in a distributed computing framework. Through markets,

local information about user preferences and resource scarcity are summarized and communicated globally in the form of prices. This information is valuable not only to agents for evaluating tradeoffs between resources, but also useful to the operating system from a more global perspective for self-tuning.

Our eventual goal is the development of computational market systems for allocating resources within distributed operating systems. A *computational market system* is an infrastructure that implements a computational economy. Economic systems, whether computational or not, are composed from a set of mechanisms and institutions for decision making as well as laws, rules, and overall organizational policy governing their implementation [11,17]. A computational market system must deploy, monitor, and adapt computational markets and institutions on an as-needed basis and in accordance with system policy. Developing computational market systems for the distributed operating systems realm has many promising applications, but also many challenges. In this paper, we content ourselves with a few exploratory case studies focusing on the potential value and uses of such systems.

1.1 Problem Description

Computer users are becoming increasingly mobile. So-called *road-warriors* travel to remote sites for hours, days, or months at a time. Workers are transferred to new departments, contractors relocate for temporary projects, and telecommuting from home is becoming increasingly common. With wireless connectivity, both users and network nodes can change location dynamically, thus changing the network topology itself.

As users relocate to different points in a distributed network, file access latency to centrally stored resources such as electronic mail and other intranet databases will vary from one location to another depending on network conditions and configuration. Most current network file management protocols and applications do not support migration of files based on user locality. Location may be keyed to a person's permanent office or PC location, or may be based on the location at time of setup. As users move to different locations on the local- or wide-area network, service times can vary drastically.

In this paper we examine the case of a set of mobile users accessing a set of geographically disperse network file servers. Users are allocated file space on one or more of the file servers. As users migrate across the system, their access latency will change depending on network configuration. In these experiments, mobile users have preferences over file access latency versus amount of file storage space. Users with a low tolerance for delay will generally be willing to trade off disk space for faster access. Users also have similar preferences

based on the types of tasks requiring these files. Thus for archived files, access latency may not be important, while for email files it is somewhat important, and for frequently used files, very important.

1.2 Related Work

There is an increasing interest in the use of markets for allocating network resources, as exemplified by the volume of material on sites providing collections of papers and projects on information and network economics [7,18,31]. While these approaches share an interest in using economic mechanisms for allocating resources, they vary widely otherwise. We consider two features, among many, to generally classify our approach.

One distinguishing feature is the resource allocation problem being addressed. Is there a single resource to allocate or are there multiple interrelated resources? Many market or market-like mechanisms for allocating computational resources of various kinds in a distributed computing environment [12,8,33,14] are centered about a global model for the resource, from which each agent or module calculates the marginal value of resource for itself. By using this value for bidding, the market allocates goods efficiently according to marginal value. More recent work on bandwidth allocation for various network or multimedia quality-of-service (QoS) requirements considers different kinds of pricing policies and problems involving dynamic resource allocation [20,29,25].

In contrast to these applications, a distributed operating system environment is concerned with finding an allocation for multiple interrelated resources. In other words, each agent is potentially interested in combinations or bundles of goods (resources, services), rather than a single type. General equilibrium theory provides a formal model for allocation of multiple interrelated resources under classical conditions such as gross substitutability, continuous goods, and complete markets. Market-oriented programming [35] implements a distributed computational economy as an instance of a general equilibrium system, and finds its competitive equilibrium. Our approach in these experiments derives from market-oriented programming. Since there are restrictive conditions on general equilibrium convergence, as well as associated time delays required for convergence, these systems generally need modification when applied to more real-time, dynamic, and discrete applications [16,5,34]. Multi-object auctions can also be used for the simultaneous sale of many resources, such as the one used by the FCC [22]. Combinatorial auctions allow bidders to bid on combinations of resources, but establishing auction properties and making them computationally-tractable is an area of current research [27,37,1].

A second distinguishing feature of market-based approaches is the design problem being addressed. Is the focus on a single system component, such as auctions [32], or is the focus on the overall system architecture and supporting middleware [15,24]? When designing a component, mechanism design and game theory [4,26] provide useful formal tools. However, as systems scale up, and the available resources or services within the system change over time, the ability to support an extensible and flexible set of mechanisms through a computational market system becomes increasingly important. Other types of computational and information networks where this kind of infrastructural, or middleware, approach have been used include digital libraries [2,9], business process management [13], and distributed databases [30].

1.3 Performance criteria

Generally, operating system performance is measured in terms of network efficiency, which involves achieving target levels of service for all users or minimizing required resources. Since users often have diverse and conflicting performance criteria, it can be difficult to define a meaningful system wide performance goal in terms of network efficiency. Using market-based methods, we can allocate disk space based both on user preferences as well as current system constraints—optimizing economic[19], rather than network, efficiency. By incorporating end-user preferences, users may get higher perceived value or utility as a result of a system's awareness of individual application needs.

Incorporation of user and application-specific preferences has useful implications for system design. It means that the system can reallocate resources flexibily and dynamically in response to a diverse and potentially changing user population. For example, if one user prefers having more disk space even if it means having slower access latency, while another user prefers faster access latency even if it means having less disk space, it will be hard to find a system-wide performance criteria that satisfies both users. Instead, if we allow the first user to acquire more of the large remote disk space while the second user acquires more of the scarce local disk space, both users may be more satisfied than if we forced them to split the remote and local disk space evenly.

While our initial experiments are based on allocating only two basic resources, the economic framework is intended for allocating multiple interrelated resources and extends to adding other operating system resources, such as signal or graphics processing, or to offering resources under different qualities of service. In such a diverse environment, finding a system-wide performance criteria based on network efficiency becomes even less tenable.

1.4 Experiment Summary

In Sections 3, 4, and 5.1, we discuss our experiments in network file allocation for mobile users and show qualitative results that arise from different economic design models and assumptions. Below is a summary of the experiments listed according to the design criteria being considered.

- *Support diversity of users and tasks:*
  In Section 3.1, 3.2, and 3.3, we consider how different parameters such as utility parameters and initial endowments can effect the final allocation. These potentially provide the operating system, or system manager, with tunable parameters for different kinds of policy-level decisions. For example, by changing the initial endowments, the operating system manager can provide some users with a greater share of system resources than others. However, evaluating the tradeoff among resources still rests with the user.
- *Adaptability:*
  In Section 3.4, we consider how price can be used by the operating system to suggest when to replicate resources. By modeling the replication technology, the system can reason about how much should be acquired. In Section 4.1, we consider how the user population's utility parameters effect the gain users get from dynamic reallocation. This kind of information could potentially be used to provide the system with feedback on how often it should reallocate resources.
- *Market vs. non-market dynamic reallocation:*
  In Section 4.2, we compare dynamic market reallocation of disk space with that of a non-market heuristic reallocation rule. Given a static and known user population, a simple heuristic rule can achieve the same optimal allocation as the market, without the overhead involved in running the market. However, when the population changes, the market mechanism still finds the optimal allocation while the heuristic rule does not—suggesting that the market approach has the potential to be more flexible and adaptable.
- *Scalability:*
  In Section 5.1, we consider the use of brokers, or information consolidators, which can both simplify the economic setup and reduce the market overhead as the number of resources and/or agents in the system are increased.

In the next section, we describe the computational economy configuration used in our experiments.

2 Designing the Mobile User Economy

To model the mobile user file allocation problem as a computational economy, we must first make explicit both the goods to be exchanged and the participating agents. We also need to specify the market mechanisms, or auctions. Below we describe these various steps in designing the mobile user economy.

2.1 Resources

Computational resources are the goods of a computational economy. Defining the available goods effectively circumscribes the design space. Our first task in designing any economy is determining what resources, or goods, will be produced and consumed.

In the mobile user economy, the resources are the available disk space across the network. To determine what system resources are available, we first need to establish the layout of the underlying network topology. In our case, we defined a simplified distributed network configuration based on the one shown in Figure 1. In Figure 1, the network configuration has three servers, each with local disk space. Co-located with each server are multiple clients. These clients have access to the entire network, but only through their local server. We refer to this combination of server, clients, and disk space as a site.

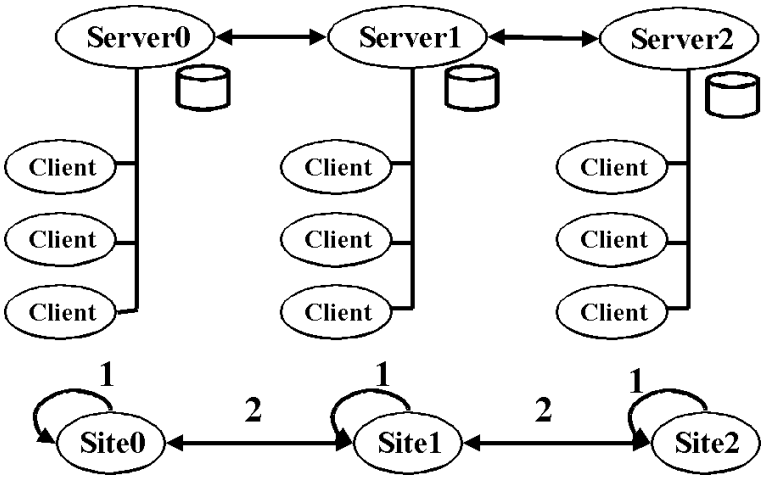

Fig. 1. A simple network configuration

The bottom of Figure 1 shows the abstract network connections for this configuration. The loop from each site to itself represents the fact that any client at a site can access the disk at that site via the local area network. The num-

bers on each link in the network represent the average access latency across that link.

For our experiments, we used the ten site network shown in Figure 2. Each link between sites has an average latency of 2 sec/Mbyte, while each site's local network has an average latency of 1 sec/Mbyte. Thus, latencies between sites are twice the latency within a site.

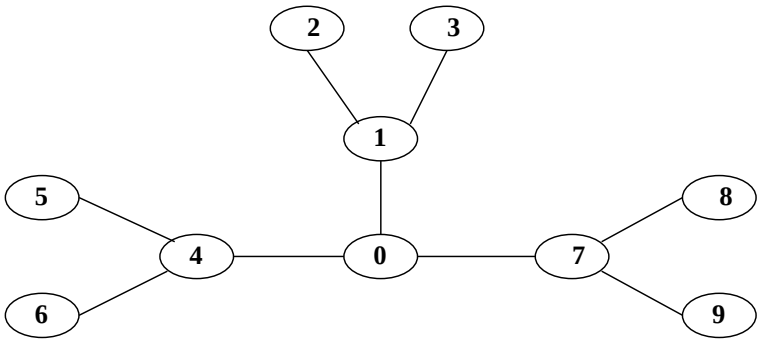

Fig. 2. Network configuration for experiments

Since disk location correlates to file access speed, users need to be able to distinguish between different disk locations. In the mobile user economy, we define disk space on the different sites to be separate goods (e.g., diskspace@site0 or diskspace@site9). In Section 5.1, we discuss how this model could be modified so that users could define their preferences in terms of disk space and access latency only.

Another important characteristic is time—most resources are actually goods over some time period. Thus the amount of disk-space must be quantified by saying over what time period it is available for—is it per hour, per day, per year? However, if the entire model operates over the same time period, it may not be necessary to explicitly put in the time period. In our case, we assume that all disk space is sold on a per-day time period.

A related question, which impacts on the choice of market mechanisms and the quality of the allocations, is whether the goods are continuous or discrete. In other words, can disk space be sold as 1.234 Mbytes or is it only available in 1 Mbyte blocks? Classical general equilibrium systems, mentioned in Section 1.2, assume that goods are continuous. With discrete goods, pareto optimality or even equilibrium is no longer guaranteed. Therefore, to simplify our economic setup for these initial case studies, we choose to consider all our resources as being continuous.

2.2 Agents: Producers and Consumers

2.2.1 Producers

Producer agents are associated with a technology, which specifies an ability to transform some goods into other goods. The sole objective of producers is to choose an activity within their technology so as to maximize profits. Since we are assuming a competitive economy, where agents act as price-takers, this amounts to supplying their goods at marginal cost.

We investigated two different types of producers, fixed capacity and variable capacity, depending on the kind of transformations we wanted to model. The fixed capacity disk storage producer has a fixed amount of disk storage space to sell. For this technology, we assume there are no incremental costs associated with heavier disk usage, so the producer marginal cost is \$0. Under this assumption, if there is more disk space available than consumers need, they will be charged \$0. It is only when disk space is scarce relative to consumer needs that the consumers will have to pay anything. This cost reflects the opportunity cost involved in using that disk space for one consumer's work over another's.

The variable capacity producer actually produces disk space with decreasing returns to scale technology. That is, the cost of incremental disk space is higher than that of existing capacity. This production technology reflects not only the cost to the system of buying new disk space but also the congestion associated with increased disk space. However, our main point was not capturing this system cost accurately, but showing how to use this configuration to ask hypothetical questions about when and where to provide additional disk space.

2.2.2 Consumers

Consumer agents are endowed with an initial quantity of goods or money and engage in trades so as to maximize their utility subject to their budget constraint. Consumers in this model represent the end-users of the distributed operating system. Consumers' behavior depends on both their utility functions and on their initial endowments. For example, given the same endowment, each user may consume different resources depending on its personal preferences. Additionally, the same user may choose different resources given different endowments or prices. By changing initial endowments, users can be given larger or smaller shares in the system wealth. Users' endowments can be in the form of some kind of system (or real) currency or might consist of owning computational resources, such as disk space or CPU on their own computer, and selling them to buy other resources.

For our experiments, a given user at a site is potentially interested in network-wide disk space given its access latency, and current prices for disk space. We considered three types of users, or disk-space consumers. The first type uses files requiring rapid-access and has twice the income of the other two consumers. This user might represent a technical user who requires more computing power. The second user type also uses rapid-access files, but with a lower income. We intended the lower income to reflect a lower system priority for this user. The third type values file space for archiving files over rapid-access, and also has a lower income than the first type. We assumed that users were willing to spread their storage allocations across different disks (or more realistically, that the system could support this kind of file distribution).

The user's utility function determines how the user makes the tradeoff between disk space at the different sites. For these experiments, we used a CES (constant elasticity of substitution) utility function, defined as follows:

$$U(x_{ds0}, x_{ds1}, x_{ds2}) = \left(\sum_{i=0}^{9} \alpha_{dsi}^{1-\rho} x_{dsi}^{\rho} \right)^{1/\rho}$$

This utility function defines the substitutability of disk space at various sites and weights them according to their access latency. The x_{dsi} refers to the quantity of disk space acquired at site i.

There are two tunable parameters, α and ρ. The coefficients α_{dsi} weight the consumer's preference for disk space from each site i. To weight faster access disk space more, we set α_{dsi} to be a function of the access latency from site i, namely $1/access\_latency^{10}$. The weighting factor 10 was simply chosen to highlight the differences in demands between rapid-access and archival users. The ρ parameter indicates the substitutability between disk space at from different sites. Obviously, disk space at one site can always be substituted for disk space at another connected site. Exactly how substitutable depends on both consumer preference and type of file access. Thus an archived file's location is quite substitutable. However, for rapid-access files, location becomes more important and disk space is less substitutable. We modeled this behavior by choosing $\rho = 0.99$ for consumer archiving and $\rho = 0.01$ for rapid-access (where $\rho = 1$ is perfect substitutability and $\rho < 0$ for complementary goods).

2.3 Auctions and Market Infrastructure

One of our aims was to provide a testbed environment to explore the effect of different kinds of auctions, agents and market configurations on resource allocation within a distributed operating system environment. To this end,

auctions were built as processes within a prototype distributed operating system. Building auctions within the simulator environment enables future work on markets that make use of system data, on measuring market overhead and performance, and on comparing market with non-market systems using real system traces.

Auctions are simply a set of rules for determining a price and/or allocation based on a bidding protocol [21] and provide a very flexible market framework—different auction types can have a large effect on the resultant resource allocation properties. Information about the different auction rules and protocols can be captured in a compact, reusable manner through the use of parameterized auction components [23,38], where auction parameters can be tuned to reflect the type of good being sold, timing requirements or mechanism properties desired. For example, disk space might be sold differently depending on how it is bundled (e.g., per-hour, per-day), characteristics (size, reliability), or to whom it is sold (individual, project). Some auction parameters include auction clearing rate (e.g., once a day, every 5 minutes, with each new bid), price setting rules (e.g., first price, second price, zero excess demand), and whether the resources are divisible or not.

Making these parameters explicit, along with formalizing the bidding interaction protocol, provides a basis for simplifying and automating both the auction creation and agent interaction process. It also provides a basis for reasoning about what kinds of auctions to create for different types of goods and/or desired allocation properties. As operating system conditions change, the system itself can apply this knowledge, changing auction parameters appropriately.

For example, for a single auction there is a tradeoff between the market clearing rate and the allocative efficiency of the result. An efficient allocation occurs when all bids have arrived and the auction sets the price where supply equals demand. However, agents may face a long delay until the resource gets allocated—for some kinds of resources this may not be important, while for others it may be critical. On the other hand, in a continuous double auction [10], bids clear as soon as a match is found. Agents face minimal delay but potentially non-efficient allocations. Intermediate market clearing rates provide different tradeoff points between agent non-delay vs. allocation efficiency. As the number of bidders increases or decreases at the auction, these tradeoff points will change. The operating system may want to adjust the auction clearing rates accordingly.

We designed the computational economy based on the market-oriented programming approach mentioned in Section 1.2. All resources are sold with continuous units, agents behave competitively (i.e., as price-takers), and auctions set prices at zero-excess demand (i.e., supply equals demand). Each auction clears whenever a new bid arrives and the system runs until it reaches equilib-

rium prices—the set of prices under which supply and demand are balanced for all the resources. Although running auctions until they reach equilibrium would not be realistic for many real-time resource allocation situations, we assumed that disk space was allocated on a per day basis so that market overhead was not an issue.

The next section describes our original market configuration, which was used in the different experimental setups in Sections 3 and 4.

2.4 Mobile User Economy Configuration

Figure 3 shows the mobile user economy configuration for the experiments in Sections 3 and 4. For simplicity, only the producers, consumers, and auctions for the first three sites are shown. A producer at site i can sell its disk space through its respective auction for disk space at site i. Consumers at any network site can bid on this disk space. The arrows represent the eventual flow of goods from producers to consumers, where auctions set the price and allocations for agents.

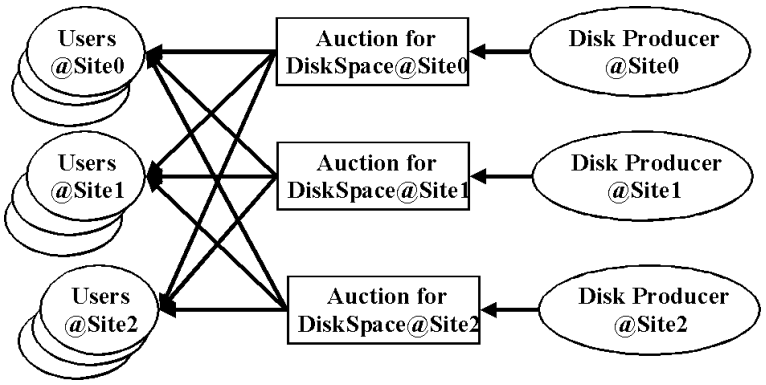

Fig. 3. Mobile user economy configuration

In the first set of scenarios in Section 3, the disk producer agents at all sites are fixed capacity producers. They own their local disk with 1000 Mbytes of space which they sell/rent each day. Since the producer's marginal cost is \$0, it has an inelastic supply bid—it will sell any amount of disk space, up to 1000 Mbytes, for \$0 (or more) dollars per Mbyte per day. In the last scenario, we use variable capacity disk producers to explore a hypothetical question about how much disk space should be provided at different locations.

3 Experiments with a Diverse User Population

In these experiments, different user types tradeoff speed of file access for amount of disk space—the actual tradeoff depending on their task and/or personal preferences. There are three different types: high-income rapid-access users, low-income rapid-access users, and low income archive users. There are a total of thirty users (ten of each of the three types) located across the ten network sites described in Section 2.1. For each scenario, we move the users and then discuss the resulting reallocation.

3.1 Scenario 1: all users reside at site 0

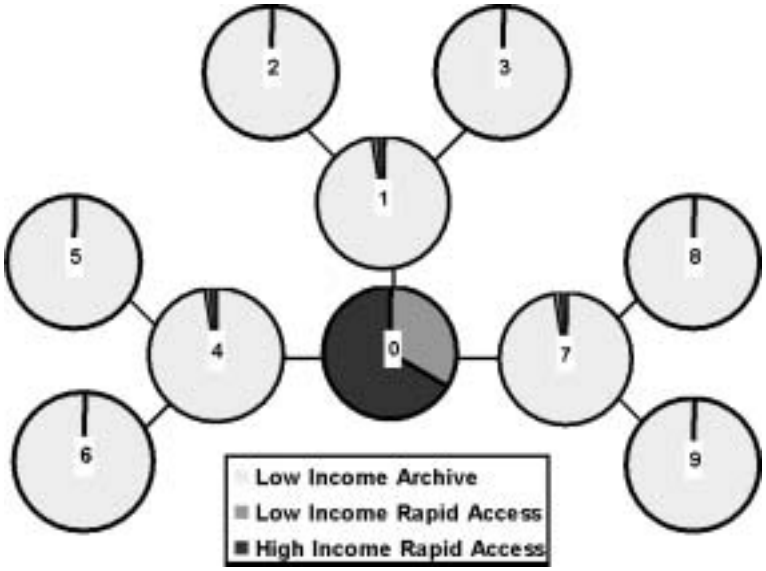

Fig. 4. Disk space allocation when all users are at Site 0

Initially all users start out at the same site, site 0. The final disk space allocations are shown in Figure 4. The amount of disk space allocated for each type of user is shown as a proportion of the total disk space at each site. Rapid access users buy almost no disk space on the most remote sites (sites 2, 3, 5, 6, 8, and 9) which have the largest latencies, and a small amount at sites 1, 4, 7, which have slightly lower latencies. Thus, the resulting allocation has rapid-access users with disk space on the nearer sites (amount depending on income), while archival users dominate the more remote sites. Since the archival users are willing to tolerate slower access rates than the rapid-access users, they are able to acquire more disk space.

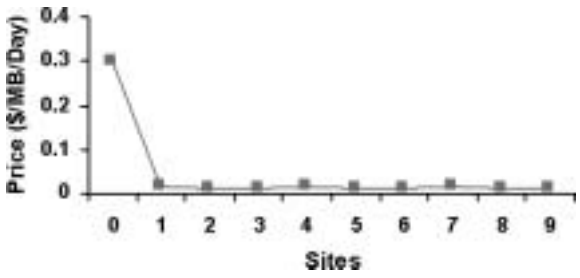

Fig. 5. Disk space prices when all users are at Site 0

Figure 5 shows the corresponding price per Mbyte/day for disk space at each site. The price of disk space at site 0 is about 30 times that at the other sites—rapid access users drive the price of disk space at site 0 up until archival users would rather buy disk space at the remote sites. Sites 1, 4, and 7 have marginally higher prices than the remaining sites since rapid access users do buy a small amount of disk space there. The congested resource is easy to spot in Figure 5.

3.2 Scenario 2: Uniform user distribution

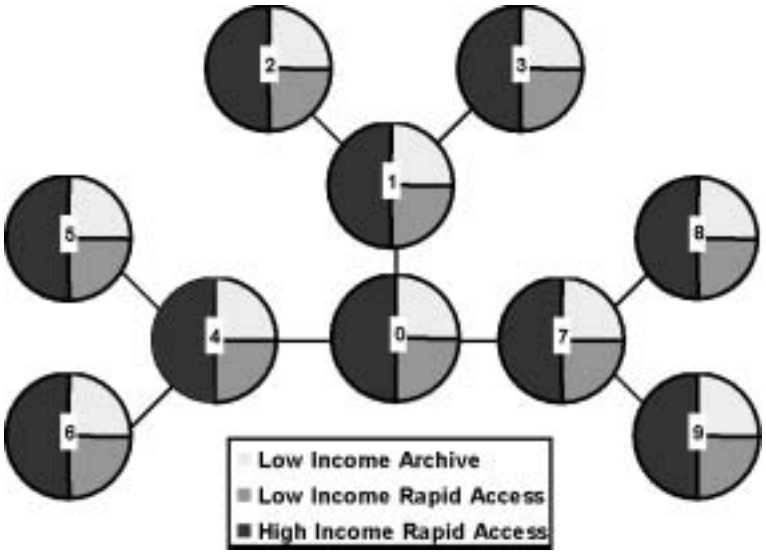

Fig. 6. Disk space allocation when users are uniformly distributed

In this scenario, the users have moved so as to be evenly distributed across the ten sites. Since demand is spread evenly between disks, it's not surprising that all prices end up being the same, namely $0.04. Given the same price, all consumers prefer local storage space to remote storage space. Thus, the amount of disk space acquired is directly proportional to income so that high-income users get twice the disk space of either of the other two user types, as

shown in Figure 6.

3.3 Scenario 3: high-income rapid-access users concentrated

Instead of a uniform distribution, suppose all of the high-income rapid-access users (e.g., technical staff) are concentrated on one sub-net of the network, say sites 2 and 3. The other two types are distributed across the remaining leaf sites—low-income rapid-access users at sites 5 and 8, low-income archival users at sites 6 and 9. Figure 7 shows rapid-access users acquiring disk space at their local sites, while archival users spread themselves among the remaining sites.

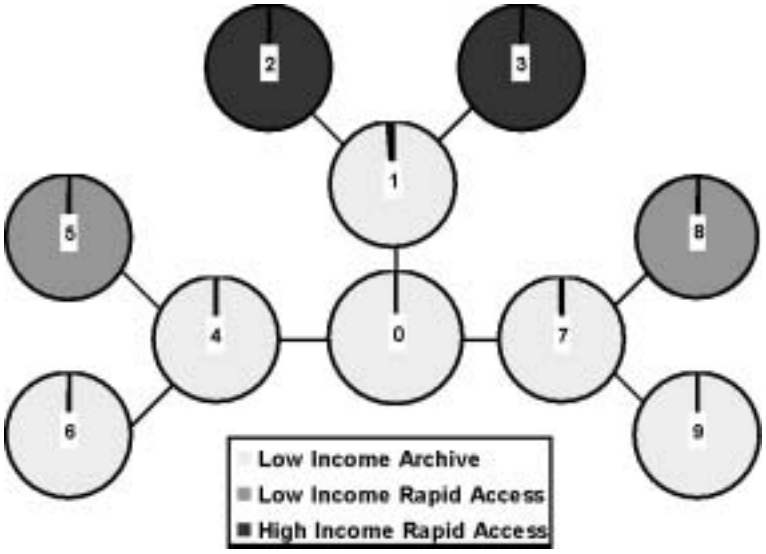

Fig. 7. Disk space allocation with high-income rapid-access users at Sites 2 and 3

In Figure 8, high prices at sites 2 and 3 reflect the high demand and willingness to pay for disk space there imposed by such a distribution. Smaller price peaks, at sites 5 and 8 represent the demand for disk space at those sites from low-income rapid-access users. In the next section, we consider how this information might be used as a basis for deciding if and how much disk space to acquire at those two sites.

3.4 Scenario 4: variable capacity producer

In scenario 3, the high prices at sites 2 and 3 reflect a high willingness to pay

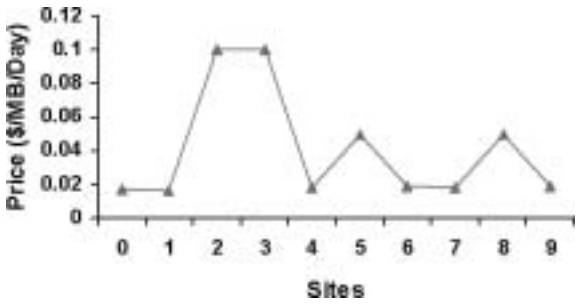

Fig. 8. Disk space prices with high-income rapid-access users at Sites 2 and 3

for disk space at those sites. A self-tuning operating system might observe the high price and consider whether to recommend acquiring more disk space at a particular location, and if so, how much. Suppose that, instead of a fixed disk size, disk space could be incrementally produced (or bought).

To explore this situation, we changed the disk producer's technology from a fixed capacity, zero-marginal-cost technology to a variable capacity, decreasing returns to scale technology. In particular, we used a quadratic-cost technology for the disk space producer at sites 2 and 3. Although the quadratic cost function is not particularly realistic, it is a representative and easily illustrated decreasing returns to scale technology. The quadratic-cost disk producer accepts money as input and produces disk space where the cost of producing more disk space goes up quadratically. The producer still supplies disk space according to its marginal cost, however its marginal cost is no longer zero. Implicit in this production technology is the additional cost required to keep access times the same across the network, in spite of having more more disk space available and therefore more file accesses.

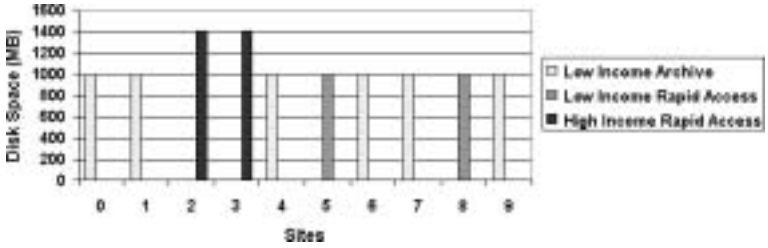

Fig. 9. Disk space allocation with variable capacity producers at Sites 2 and 3

Since we were not modeling any real costs or technology here, we calibrated the parameters of the production function so that a variable capacity producer could supply 1000 MBytes for half the price of disk space at sites 2 and 3 in scenario 3. Although this choice is somewhat arbitrary, the general idea is to consider what happens when the marginal cost of producing more disk space at sites 2 and 3 is cheaper than its current price. Comparing the final allocations in Figure 9 to those in scenario 3, we see that sites 2 and 3 (location of the high income, rapid-access consumers) now support 1400 MBytes of disk at these sites, while the other allocations remain essentially the same.

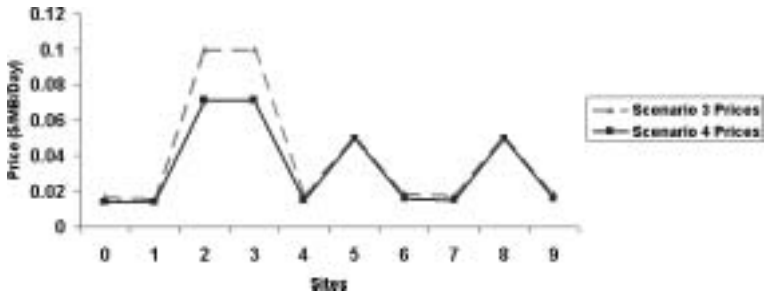

Fig. 10. Disk space prices for Scenario 3 and Scenario 4 compared

Figure 10 compares the final prices, given the increased disk space, to the original prices in scenario 3. The prices for disk space at sites 2 and 3 have been reduced; users are no longer willing to pay more for this disk space than the incremental cost of acquiring new disk space. This scenario suggests how price information could potentially be used as the basis for acquiring new disk space and/or determining where more disk space should be located.

4 Mobility Experiments

Having explored the effects of having diverse types of mobile users in the previous section, we now look at the effects of mobility alone on the users' welfare. First, we show how users' substitutability affects the gain they receive from a dynamic reallocation and how the system might use this information. Second, we compare the market reallocation mechanism with both the original static allocation and a dynamic reallocation rule.

4.1 How substitutability affects reallocation value

To eliminate the diversity aspect, we used 100 users all with the same endowments and same ρ and αs. Initially users were distributed evenly across the network, each with a static allocation of 100 Mbytes of local disk space. Next, we randomly moved some percentage of the population to a random location within the network and ran the economy to get a new set of disk space allocations. We then measured the utility gain or loss for each agent between its current utility and the utility of its original static allocation. This utility gain measures the benefits of pricing over fixed static allocation. Since individual values utility gains or losses varied, we measured average utility gain per user. We found the benefits of flexible pricing are proportional to the

overall mobility of the population, as might be expected. However, the other major determinant of net gain is the degree of substitutability of resources in the economy.

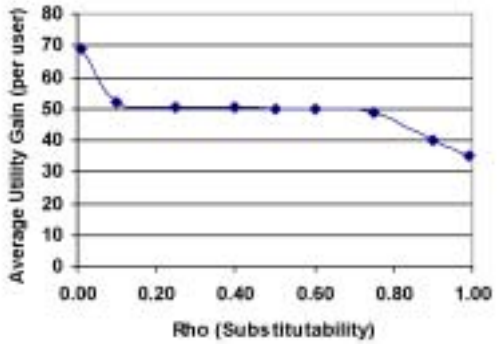

Fig. 11. Average Utility Gain vs. Substitutability for a mobile user.

In Figure 11, we consider how ρ—that is, how substitutable the users consider system resources—effects the utility gain. On the x-axis, larger ρ values correspond to more substitutable goods. This graph shows the effects when fifty percent of the population moves, however the results for other mobility percentages have a similar curves. We see that users with less substitutability get larger utility from reallocating disk space dynamically. This kind of information might be used to gauge how often system resources need to be reallocated—presumably when mobility is less and resources are more substitutable, the system might want to reallocate resources more infrequently than when mobility is high and goods less substitutable.

4.2 Comparing reallocation mechanisms

While on average mobile users experienced a utility gain with the market reallocation of disk space above, this presumably could be demonstrated with any reasonable dynamic reallocation mechanism. In this section, we compare market reallocation to a non-market dynamic reallocation rule which caters to rapid-access users.

We assume that the network configuration has users only at the leaf sites (i.e., sites 2, 3, 5, 6, 7 and 8). Our non-market dynamic reallocation rule assigns disk space evenly based on the following locality constraints. First, disk space on a leaf site is distributed among users at that site. Second, disk space on an internal site is distributed among users below it in the network tree. Thus, disk space at site 1 is distributed evenly among users at sites 2 and 3, while disk space at site 0 is distributed evenly among users at all the leaf sites.

Users were initially distributed uniformly among the leaf sites (96 users, 16 per site) and then a percentage of the population was moved randomly to another leaf site. This rule performs as well as the market rule when all the users are rapid-access users, and without the overhead incurred by using the market mechanism. Both, not surprisingly, outperform the original static allocation.

However, when the user population consists of half rapid-access users and half archival users, the market reallocation has higher utility gains than the non-market one. In Figure 12, we show the results when the user population is split evenly between rapid-access and archival users. Initially, users were distributed evenly among the leaf sites. We then randomly moved different percentages of the user population and compared the market and non-market reallocations.

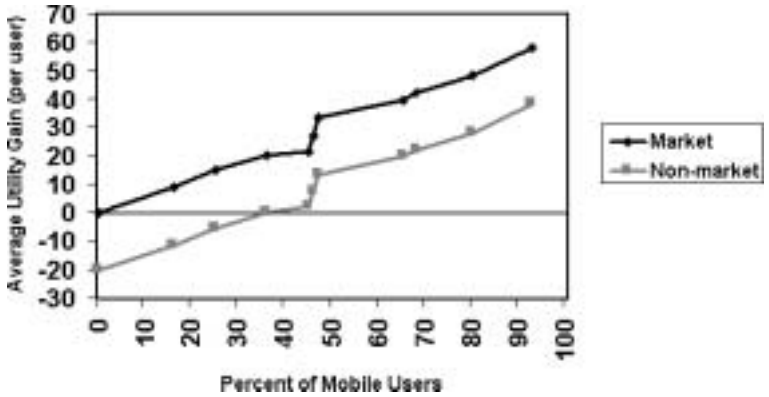

Fig. 12. Dynamic reallocation: market vs. non-market

As the users move, the average utility achieved with no reallocation is used as the baseline, and considered as having zero utility gain in Figure 12. Average utility gains for the two dynamic reallocation methods are always measured relative to those of the static allocation. We do not currently know if the non-linearity around fifty percent mobile has any significance, or is just an artifact of the particular network configuration or user population distribution. Elsewhere the benefits of both market and non-market reallocations are linear with respect to the mobility of the population. However, the non-market reallocation does noticeably worse than the market one with a more diverse population than the one it was designed for.

We recognize that this is a very simple non-market reallocation rule and more sophisticated, adaptive rules could conceivably give better results. However, while a market allocation is not necessarily going to be better than a non-market one for a fixed scenario, we would argue that it is much more flexible and scalable—because it incorporates user preferences in a principled way, the benefit of the economic solution is its ability to dynamically respond to changes in the user population. Presumably the more dynamic and large scale the environment, the harder it will be to find a good adaptive reallocation rule.

A natural extension of this work is to compare currently used reallocation mechanisms to market ones, not only the allocative efficiency but also how much overhead each approach adds. All of these issues (and more) remain open for future investigations.

5 Market Design Considerations

5.1 Scalability: brokers

Although the economic model in the previous sections displays reasonable qualitative behavior, it has two disadvantages. First, every consumer has to interact with every auction. Second, every consumer has to know about the access latency between their local site and all others in the network. Both of these factors make it hard to scale up the design for large-scale networks. In this section, we consider how the use of brokers helps consolidate some of the access latency information requirements and also hide the complexity of the system from the users.

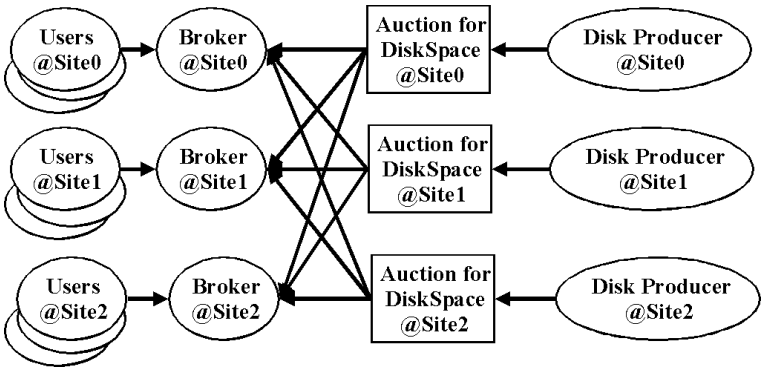

Fig. 13. Mobile User Economy with brokers mediating information flow for auctions.

Brokers, as shown in Figure 13, reduce the network monitoring costs of gathering information about average access latency by each individual agent. Having a broker gather this information once per site reduces system overhead and application complexity. Suppose that there are n_a agents and n_s sites, and assuming that it costs 2 messages to actively measure network latencies to any site, then it will cost $2n_s n_a$ messages for all the agents to individually acquire this information. However, assuming the typical case where $n_a \gg n_s$, then by using brokers it will only cost $2n_s^2$ messages. In the worst case, where $n_s \gg n_a$,

it will never cost more than not having brokers, since brokers will only exist on sites with agents. Brokers might also passively monitor and measure the latency of bidding or other traffic, potentially reducing the extra network traffic to zero. Another transaction cost improvement includes consolidating bids from a given site to both reduce network traffic and cut down on auction processing time (since it only has to handle one bid instead of n). Finally, auctions themselves could be optimized for particular utility or bidding forms [39].

Brokers can also help reduce agent decision complexity costs. Since many applications will tend to have stereotypical values, a broker can provide different quality of service mappings [28] per application. For example, if email is classified as a rapid-access application, then its ρ parameter could be automatically set to 0.01 unless overridden.

Finally, application and system constraints regarding minimum disk space size can be monitored and enforced by the broker. The broker may also be designed to provide additional value-added processing by using historical information about access latency or predictive rates. Designing the broker in this manner doesn't effect the final resource allocation of the system. However, modularizing the network-aware software into the broker can reduce transaction and agent decision costs, as well as provide enforcement of system policies or other value-added local information services.

5.2 Future Work

5.2.1 Simplify agent good space

If the economy was extended to consider network transportation as an explicit good, then brokers could be used to simplify the user's decision about where to acquire disk space. Instead of users having to specify their preferences for storage at every disk site, brokers would allow them to define their preferences in terms of access latency and disk space only. We describe below how a broker could replicates a user's utility function once per available disk and combines them using application-specific or system-specific utility parameters into a new nested utility function. For example, let's define the user's utility function between disk space, x_{ds}, and access latency, x_{al}, as $U(x_{ds}, x_{al})$

Assuming the broker determines the application-specific CES parameters to be α and γ, and there are available disks at sites A and B, then the broker's utility function is

$$U(x_{dsA}, x_{alA}, x_{dsB}, x_{alB}) = \left(\alpha_{dsA}^{1-\gamma} U_{dsA}(x_{dsA}, x_{alA})^{\gamma}\right.$$

$$+\alpha_{dsB}^{1-\gamma} U_{dsB}(x_{dsB}, x_{alB})^{\gamma}\Big)^{1/\gamma}$$

where U_{dsA} and U_{dsB} are copies of the user's original utility function.

From this composite function, one can analytically derive the appropriate four demand functions for x_{dsA}, x_{alA}, x_{dsB}, and x_{alB}. However, the users only need to know their own preferences between disk space and latency. Additionally, once network transportation is priced, a natural extension would be for brokers to own the network links and then provide admission control [36].

5.2.2 Transacting out of equilibrium

Another assumption made in these experiments was that disk space was allocated statically once a day, after all the markets had converged to an equilibrium, as opposed to dynamically, as needed. If we relax this assumption, a continuous double auction would allow the agents to buy disk space immediately. Of course, agents buying disk space earlier may end up paying less than agents who buy later. Also, there may be some agents willing to pay more for disk space than agents currently using it, who would be rejected—clearly not pareto efficient. If instead, we use a second price auction, all files will priced at the highest amount of the denied files, as in the internet smart market approach [20]. Files would get bumped by higher paying, and therefore higher valued files, and have to relocate to more inexpensive disk space. Although this policy is optimal for a single disk space auction, we have not seriously addressed the implications for multiple disk space auctions.

6 Conclusion

In this paper we have formulated a market economy for distributed file allocation for mobile users. Results show that a market based solution based on a notion of economic, rather than network, efficiency can result in solutions that increase overall welfare of users when compared to fixed allocations. These mechanisms are flexible with respect to user mobility and diversity—as users move, or user populations change, prices adjust so that consumers demand just the amount of disk space available in accordance with their utilities.

Adjusting endowments provides additional control of the distributed file system. In the case of the high and low income rapid-access users, we used initial endowments to provide a means of adjusting priorities across users. One can imagine endowments being determined externally, as part of the consumers overall optimization for file allocation versus all other goods. This scheme could form the basis of file allocation scheme based on willingness-to-pay.

We also examined how the introduction of variable capacity producers of disk space makes it possible to reason about acquiring or migrating disk space in response to mobile demands. Finally, we examined a number of issues related to market design. Use of brokers reduces communication overhead by not requiring individual users to have knowledge of network latencies and disk locations, however there may be some loss in flexibility in terms of price stability and efficiency of allocations.

Extending this framework to deal with a real-world distributed operating system environment requires relaxing many of the simplifying assumptions made, such as requiring continuous goods or transacting only upon equilibrium. We would like to pursue some of these issues as well as extend the simulation environment by adding resources (including resources with quality of service) and using a more complex network topology. Although the results presented here provide some indication of the applicability of auction-based market mechanisms to distributed file system operations, additional analysis is needed to determine the scalability of the approach and to quantify the benefits relative to both standard fixed allocation and dynamic reallocation approaches.

Acknowledgements

We would like to thank Eric Horvitz and John Doceur at Microsoft Research who contributed to the concepts and work described here.

References

[1] Sushil Bikhchandani and Joseph M. Ostroy. The package assignment model. In *Maryland Auction Conference*, May 1998.

[2] William P. Birmingham, Karen M. Drabenstott, Carolyn O. Frost, Amy J. Warner, and Katherine Willis. The University of Michigan Digital Library: This is not your father's library. In *Proceedings of Digital Libraries '94*, pages 53–60, June 1994.

[3] William J. Bolosky, Richard P. Draves, Robert P. Fitzgerald, Christopher W. Fraser, Michael B. Jones, Todd B. Knoblock, and Rick Rashid. Operating system directions for the next millennium. Technical report, Microsoft Research. http://research.microsoft.com/os/.

[4] Donald E. Campbell. *Resource Allocation Mechanisms*. Cambridge University Press, 1987.

[5] John Q. Cheng. *Essays on designing economic mechanisms*. PhD thesis, University of Michigan, Economics Department, 1998.

[6] Scott H. Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1996.

[7] Nicholas Economides. The economics of networks. http://raven.stern.nyu.edu/networks/biblio_hframe.html.

[8] Donald F. Ferguson, Christos Nikolaou, Jakka Sairamesh, and Yechiam Yemini. Economic models for allocating resources in computer systems. In Clearwater [6].

[9] Innes A. Ferguson and Michael J. Wooldridge. Paying their way: commercial digital libraries for the 21st century. *D-Lib Magazine*, June 1997.

[10] Daniel Friedman and John Rust, editors. *The Double Auction Market*. Addison-Wesley, 1993.

[11] Paul R. Gregory and Robert C. Stuart. *Comparative Economic Systems*. Houghton Mifflin Company, 1989.

[12] Kieran Harty and David Cheriton. A market approach to operating system memory allocation. In Clearwater [6].

[13] N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 2 & 3(5):105–130, 1996.

[14] James F. Kurose and Rahul Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, May 1989.

[15] Spyros Lalis, Christos Nikolaou, Dimitris Papadakis, and Manolis Marazakis. Market-driven service allocation in a qos-capable environment. In *First International Conference on Information and Computation Economies*, pages 92–100, October 1998.

[16] Aurel A. Lazar and Nemo Semret. Design, analysis and simulation of the progressive second price auction for network bandwidth sharing. Technical Report 487-98-21, Department of Electrical Engineering, Columbia University, 1998.

[17] Jeffrey K. MacKie-Mason. Multi-agent systems, mechanism design, and institutions. Invited talk handout at The First International Conference on Information and Computation Economies (ICE-98).

[18] Jeffrey K. MacKie-Mason. Telecom information resources on the internet. http://china.si.umich.edu/telecom/telecom-info.html.

[19] Jeffrey K. MacKie-Mason, Scott Shenker, and Hal R. Varian. Service architecture and content provision: the network provider as editor. *Telecommunications Policy*, 20(3):203–217, April 1996.

[20] Jeffrey K. MacKie-Mason and Hal R. Varian. Pricing congestible resources. *IEEE Journal on Selected Areas in Communications*, 13(7):1141–1149, 1995.

[21] R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

[22] John McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8:145–162, 1994.

[23] Tracy Mullen and Michael P. Wellman. Market-based negotiation for digital library services. In *Second USENIX Workshop on Electronic Commerce*, November 1996.

[24] Ori Regev and Noam Nisan. The popcorn market - an online market for computational resources. In *First International Conference on Information and Computation Economies*, pages 148–157, October 1998.

[25] D. Reininger, D. Raychaudhuri, and M. Ott. Market based bandwidth allocation policies for qos control in broadband networks. In *First International Conference on Information and Computation Economies*, pages 101–110, October 1998.

[26] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

[27] Michael H. Rothkopf, Aleksandar Pekec, and Ronald M. Harstad. Computationally manageable combinatorial auctions. Technical Report RRR 13-95, RUTCOR, Rutgers University, April 1995.

[28] J. Sairamesh, D. Ferguson, and Yemini Y. An approach to pricing, optimal allocations, and quality of service provisioning in high-speed networks. In *Proceedings of the INFOCOM'95*, 1995.

[29] S. Shenker, D. Clark, D. Estrin, and S. Herzog. Pricing in computer network: Reshaping the research agenda. *Communications Policy*, 20(1), 1996.

[30] Michael Stonebraker, Aoki M., Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, and Andrew Yu. Mariposa: A wide-area distributed database system. *VLDB*, 5:48–63, January 1996.

[31] Hal R. Varian. The information economy. http://www.sims.berkeley.edu/resources/infoecon/.

[32] Hal R. Varian. Economic mechanism design for computerized agents. In *First USENIX Workshop on Electronic Commerce*, July 1995.

[33] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18:103–117, 1992.

[34] W. E. Walsh, M. P. Wellman, P. R. Wurman, and MacKie-Mason J. K. Some economics of market-based distributed scheduling. In *Eighteenth International Conference on Distributed Computing Systems (ICDCS-98)*, May 1998.

[35] Michael P. Wellman. A computational market model for distributed configuration design. *AI for Engineering, Design, and Manufacturing (AI EDAM)*, 9:125–133, 1995.

[36] Michael P. Wellman, Jeffrey K. MacKie-Mason, and Sugih Jamin. Market-based adaptive architectures for information survivability. http://www-personal.umich.edu/~jmm/papers/darpa/Surv_Proposal.html, 1997.

[37] Peter R. Wurman. Multidimensional auction design for computational economies: A dissertation proposal. http://ai.eecs.umich.edu/people/pwurman/work.html.

[38] Peter R. Wurman, Michael P. Wellman, and William E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Second International Conference on Autonomous Agents*, May 1998.

[39] Fredrik Ygge and Hans Akkermans. Power load management as a computational market. In *Proceedings of International Conference on Multi-Agent Systems*, pages 393–400, 1996.