Market-based Resource Allocation in Grids

B. Pourebrahimi, K. Bertels, G. M. Kandru, S. Vassiliadis
Delft University of Technology
Computer Engineering Laboratory
{behnaz, koen, mahesh, stamatis} @ce.et.tudelft.nl

Abstract

The core goal of resource management is to establish a mutual agreement between a resource producer and a resource consumer by which the provider agrees to supply a capability that can be used to perform some tasks on behalf of the consumer. Market-based approaches introduce money and pricing as the technique for coordination between consumers and producers of resources. In this paper, we propose a market-based mechanism to allocate computational resources (CPU time) with a single central Market in a local Grid. In such a network whenever any node can offer idle CPU time to the Grid and whenever a node has some tasks waiting for free CPU, it may request the resource from the Grid. In our approach, consumers and producers are autonomous agents that make their own decisions according to their capabilities and their local knowledge. Continuous Double Auction model is used as a technique using which these selfish agents can coordinate their work and make their decision. The performance of this mechanism is evaluated and is compared with the simple FCFS mechanism.

1 Introduction

Decentralized computing systems are becoming increasingly popular as they enable organizations to use existing computing resources that otherwise lie idle. Whether this paradigm will be successful largely depends on the flexibility and easiness with which it can be implemented and managed.

The research presented in this paper targets heterogeneous, ad hoc Grids that could be deployed in any organization having a LAN with any number of computers. The basic idea is to process tasks on any of those machines, whenever their resources are available. Such a setting poses some specific challenges as the resources are geographically distributed, heterogeneous in nature, owned by different individuals or organizations, have different access and cost models, and have dynamically varying loads and availability.

Conventional resource management schemes are based on relatively static models when a centralized controller manages jobs and resources. Indeed, they focus on efficient allocation schedules which can optimize a given performance metric such as allocation time, resource utilization or system throughput and these management strategies might work well where resources are known in advance. However, this fails to work in heterogeneous and dynamic systems where jobs need to be executed by computing resources whose availability is difficult to predict. Where centralized approaches show some evident limitations, a completely decentralized approach also poses specific problems. Where it seems rational that each node can decide whether it needs additional resources or on the contrary wants to sell them, the main challenge is to make sure they find the resources needed or a user of their resources. One way to provide such a facility is to use a market-based approach. In this way decentralization is provided by distributing the decision-making process across all users and resource owners. Even though this approach is not novel, we intend to use it as a way of obtaining self-organization. The price of the resources reflects the need for them. If the price is high for a particular resource, the system should re-organize itself as to increase the supply of this resource. Such re-organization can be seen for instance in the context of QoS when certain service levels have to be ensured.

The main contribution of the paper is to provide a deeper understanding of the choices one

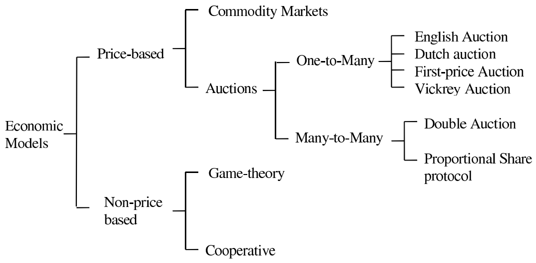

Figure 1. Economic Models.

can make as far as economic approaches for resource allocation is concerned. We present a simple yet powerful approach where individual decision making leads, through the mediation of a centralized mechanism, to a particular system behavior in which a large amount of decentralized information is condensed into a single, simple entity, namely the price. We study the impact of using a market-based resource allocation mechanism in a local Grid and compare it to a simple but high throughput alternative. The idea is to have a collection of agents as consumers and producers that interact through some central instance. Agents are modeled as buyers and sellers of tasks and resources, willing to spend money in order to fulfill their objectives. The auctioneer is another type of agent, acting as a mediator between the consumer and producer agents.

The paper is structured as follows: in section 2, we give an overview of related works in economic based resource management. Section 3 discusses our proposed approach and explains the system architecture and presents the pricing algorithm. Experimental results are shown in section 4. Finally we conclude by discussing future research.

2 Related Work

There are two main economic methods to allocate resources among the competing agents (Figure 1). One of them is the non-price based economy and the other is the price based economy. The non-pricing approaches [4] are either game theory-based or cooperative mechanisms. Game theory [12] is grounded on principles such as selfish optimization and individual utility functions. Cooperative methods have a global utility function which is known to all nodes in the distributed system. In these approaches, each agent is initially endowed with some resources. They exchange them until the marginal rate of substitution of the resources is the same for all the agents. At this point, there is no further incentive for cooperation. An example of this approach is [6] that uses decentralized algorithms to allocate resources (such as files or file fragments) in a cooperative method.

Price-based approaches are classified into two main categories of economic models: Auctions and Commodity Markets. Wolski et al [16] have used the commodity market approach to allocate two types of resources (CPU and disk storage) in grid. Allocations are done based on reaching an equilibrium price where demand equals the supply.

The auction protocols are either one-to-many or many-to-many. In one-to-many auctions one agent initiates an auction and a number of other agents can make a bid. The English auction, Dutch auction, first-price auction, second-price (Vickrey auction) belong to this category. The basic philosophy behind these auctions is that the highest bidder always gets the resource and the current price for a resource is determined by the bid prices. Popcorn [9] and Spawn [13] are examples of this approach. In many-to-many auctions, several agents initiate an auction and several other agents can bid in the auction. The double auction is the most widely used auction protocol for many-to-many auctions. In these auctions, buyers and sellers are treated symmetrically with buyers submitting bids and sellers submitting offers. There are two types of double auctions, continuous double auction (CDA) and periodic double auction. Continuous Double Auction matches buyers and sellers immediately on detection of compatible bids. A periodic version of the double auction instead collects bids over a specified interval of time, then clears the market at the expiration of the bidding interval [17]. JaWS [8], [11] and [10] are examples which use double auction model.

The Proportional Share Protocol (PSP) is a similar protocol to Continuous Double Auction, as both use a centralized scheduling algorithm. In a PSP several tasks can execute on a server at a time. The amount of resources allocated to a task depends on its price bid in relation to the sum of price bids of all tasks executing on that server. Proportional Share Protocol is proposed for the scheduling of tasks in computational clusters [14].

Although economic models have been used widely in resource allocation algorithms [15] [1], the question about which model is the most appropriate for resource allocation in grid, is not fully addressed yet. In the literature, we can find several studies on auction based resource allocation. In

[3], three types of auction allocation protocols are investigated; First-Price Auction, Vickrey Auction and Double Auction. The protocols are compared from producer and consumer perspectives in terms of resource utilization, resource profit and user payment. The experiments are characterized by a limited number of resources with predefined capabilities as well as the use of a reservation price. They distinguish between two categories of users; Risk Averse Users and Risk Neutral Users. Their results shows that the First-Price Auction is better from the consumer's perspective while Vickrey Auction is better from producer's perspective. The Double Auction favors both producers and consumers. [5] compares three different Double-Auction protocol from both the producer and consumer perspectives in terms of resource utilization, resource profit and spent budget. It concludes that Continuous Double Auction (CDA) protocol performs best from both perspectives. Assuncao et al. [2] investigate the communication demand or complexity of auction protocols in a Grid environment. Their experiments show that the English auction has higher communication requirements while CDA requires the lowest number of communications.

What distinguishes our work from the others is the use of a dynamic pricing strategy. In our model, consumers and producers propose a price based on their past experiences and current needs. The simulation is done in a larger Grid with a set of heterogeneous resources and tasks. In the current simulation, no budget limitation is considered.

3 Proposed Market-based Model

We have implemented Continuous Double-Auction mechanism as an economic model to allocate resources in a local grid. In this model, an auctioneer acts as a mediator between buyers and sellers. In a Continuous Double Auction, buy orders (requests) and sell orders (offers) may be submitted at anytime during the trading period. The users and providers of resources put their requests or offers attached with a price into the resource market as bids. If at any time there are open requests and offers that match or are compatible in terms of price and requirements (e.g. quantity of resources), a trade is executed immediately. This contrasts with other approach where a general equilibrium price computed at which the market clears.

We employ multi-agent systems as a platform for our system. Unlike centralized approaches that have a scheduler that plans schedulers for all tasks and resources, the agents in this approach plan for their own task and resources. Partitioning the system into a collection of agents can dramatically reduce the complexity by converting a complex global allocation problem into a set of smaller, independent problems. In the following section, we discuss the overall system organization and present the pricing functions by which buyers and sellers compute their prices and propose a transaction.

3.1 System Architecture

The system is composed of three entities: Buyer, Seller and Auctioneer. The market works in the following simple manner: the buyers and sellers agents announce their desire to buy or sell processing power to the market. In the Continuous Double Auction, the market acts as an auctioneer and finds the matches between buyers and sellers by matching offers (starting with lowest price and moving up) with demand bids (starting with highest price and moving down). When a task query arrives at the market place, the protocol searches all available resource offers and returns the best match which satisfies the task's constraints (such as resource size, time frame and price). If no match is found, the task query object is stored in a queue. The queries are kept in the queue till the time to live (TTL) has expired or a match has been found. When a resource becomes available and several tasks are waiting, the one with the highest price bid is processed first.

The system components can be summarized as follows:

- **Buyer/Seller Agent**: There is one buyer/seller agent per node. A buyer/seller agent controls the process of buying/selling resources by estimating the execution time of the job or availability of the resource, calculating the price and generating and submitting a request/offer for corresponding job/resource. Submitting/accepting the job to the matched seller/buyer is also the task of these agents.

- **Auctioneer Agent**: The auctioneer agent controls the market using a double auction protocol. Based on this protocol, every seller and buyer sends its offers and requests to the auction. The auctioneer inserts each received request or offer in its sorted depositories. The

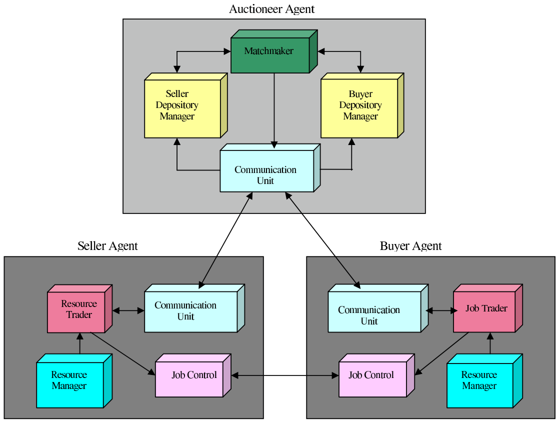

Figure 2. System agents components.

requests are sorted from high price to low price and the offers are sorted from low price to high price. A request is matched with an offer if the resource offered by the producer meets the consumer requirements regarding the quantity, time and price.

3.1.1 Agent Attributes

Every agent consists of some components as depicted in Figure 2, some of them are common among agents and some are special for a particular agent. The component that is common for all three agents, is:

- **Communication unit**: is responsible for exchanging the messages between buyers, sellers and auctioneer. TCP/IP protocol is used for communication between the agents.

The common attributes between buyer and seller agents are:

- **Resource manager**: decides whether a node needs additional resources or it has free resources. The decision is made with considering the job queue and available resources on the node.

- **Job controller**: is responsible for transferring jobs between buyer and seller nodes. Shepherding the job through the system, accept, deploy and launch the job is controlled by this unit.

whether the agent is buyer or seller, the only special attribute of the buyer/seller agent is:

- **Job/Resource Trader**: decides about the price that buyer/seller offers for resource.

The market agent consists of these components:

- **Buyer/Seller Depository Manager**: is responsible for managing all requests/offers received from buyer/seller agents. Updating, sorting, inserting the messages, and deleting the expired ones are the operations that are done by this unit.

- **Matchmaker**: It is the core unit of auctioneer that finds the matched pairs from buyer and seller depositories.

3.2 Pricing Algorithm

In a price based system, the resources are priced based on the demand, supply, and the wealth in the economic system. The prices vary with the demand and supply of the resources. In each market, the objective of a seller is to maximize its earning as much as possible and the objective of a buyer is to spend money as little as possible. Based on these objectives, the strategy of producers of resources is to raise the price when the demand for associated resource is high and lower the price when the demand is low. On the other hand, the strategy of consumers of the resources is to lower the price when supply is high and raise the price when the supply is low. Based on these strategies, we define the seller and buyer pricing function as follows:

Sellers or buyers start with a fixed price and update it over time.

$$p(t) = p(t-1) + \Delta p \qquad (1)$$

The value of Δp determines whether the price is increasing or decreasing. To change the price according to the demand or supply in the system, Δp is defined based on the past resource or task utilization on this particular seller/buyer. The following equation is an extention of the model described in [7]. Δp for seller and buyer is calculated as below: for seller:

$$\Delta p = \alpha(u(t) - u_{th})p(t-1) \qquad (2)$$

for buyer:

$$\Delta p = \beta(u_{th} - u(t))p(t-1) \qquad (3)$$

where $u(t)$ is resource/task utilization at the individual node and u_{th} is a threshold below which, the resource/task utilization should not go. u_{th} could be interpreted as the degree of **laziness** of the agent. If it is very low, it implies that the agent is satisfied with a low usage of his resources or a low completion rate of his tasks. If it is high, the agent is more demanding for himself by imposing higher satisfaction thresholds. α and β are the coefficients that control the rate of price changing. $u(t)$ is defined as:

$$u(t) = \sum_{i=t_0}^{t} x(i) / \sum_{i=t_0}^{t} N(i) \qquad (4)$$

Where $\sum_{i=t_0}^{t} x(i)$ is the total numbers of sold/purchased resources in the time period $[t_0, t]$ and $\sum_{i=t_0}^{t} N(t)$ is the total numbers of offered/requested resources in the time period $[t_0, t]$.

The sellers and buyers submit their price along with the quantity of requested or offered resources to the auctioneer. The auctioneer finds the matched pairs and the trade between each pair is made at the average of the corresponding buyer's and seller's prices.

4 Experimental Model

In order to simulate our experiments, we have implemented a Java-based platform which is used as a test-bed to simulate the mechanism with varying parameters. We set up a Grid like environment based on a local LAN in which our application test-bed is developed using J2EE and Enterprise Java beans. A JBOSS application server is used to implement the auctioneer. This server continuously receives offers and requests messages from clients (producers and consumers). Whenever a match is found by the auctioneer, it informs the respective consumer and producer by sending a message to them. Java Message Service (JMS) is used for the communication between clients and auctioneer. A predefined format of message is used for all exchanged messages between nodes. MySQL server is used as a database server to store the results of our simulation.

The network consists of N agents. Some of these agents, called consumers have tasks to perform for which they need additional resources and some, called producers have idle resources to offer. CPU time is considered as the resource in our system. Whenever a consumer needs additional CPU time for running a job, it sends a request to the auctioneer and whenever a producer has some idle CPU time, it sends an offer. In the sent messages, in addition to resource quantity and price, a Time To Live (TTL) is also included in the message. It represents the time during which the request or offer is valid or available. The simulation is done in an environment with 40 nodes with various CPU speeds. Each node creates a number of requests or offers during the simulation time. For each request the resource requirements are expressed in terms of job execution time that is generated randomly in a specific range. The offered resource, the time during which a CPU is idle, is generated randomly as well. As nodes have different processors with different CPU speeds, this heterogeneity is taken into account for matching resource requirements and offers.

Network Condition	No. of Offers	No. of Requests	No. of Matches	Resource Utilization	Task Utilization
Resources $\simeq$ Tasks	1033	967	856	83%	88%
Resources $>>$ Tasks	1627	373	363	22%	97%
Resources $<<$ Tasks	324	1111	320	99%	29%

Table 1. Continuous Double Auction Algorithm.

In this work, we perform the experiment on three different kinds of network conditions: **the balanced network** which is the type of the network where there is more or less an equal number of tasks and resources, **the task intensive network** where there are more tasks than resources and **the resource intensive network** where there are more resources than tasks.

The behavior of the price is studied in these three network conditions and the efficiency of the system is compared with a blind and simple matchmaking algorithm. The efficiency of the system is measured in the terms of task and resource utilization. Task utilization is the ratio of allocated tasks to all sent resource requests and in the same way resource utilization is the ratio of allocated resources to all sent resource offers. We also investigate the impact of the varying parameters in pricing function on the system efficiency.

4.1 Pricing Behavior

In this section, we study the evolution of the transaction price given the three network conditions. Buyers and sellers start with a fixed price and then update the price based on the demand and supply over time. When a match is found, the trade is made at the average value of the buyer and seller price.

In the current experiment, the following parameters values have been considered : $u_{th} = 0.9$ and $\alpha = \beta = 0.25$ and a price value of 4 is as a starting point. The tasks and resources are generated randomly with the probability of 50%-50% for balanced network, 20%-80% for resource intensive network and 80%-20% for task intensive network.

4.1.1 Balanced Network

The first experiment looks at a balanced market in which the supply equals the demand. In such a market, we do not expect to see any up or downward trend. As it can be observed from figure 3, the price indeed fluctuates around a level of 4 which we have chosen as the starting price value for all agents.

Looking at table 1, task and resource utilization of 88% and 83% shows the balance between supply and demand in such network.

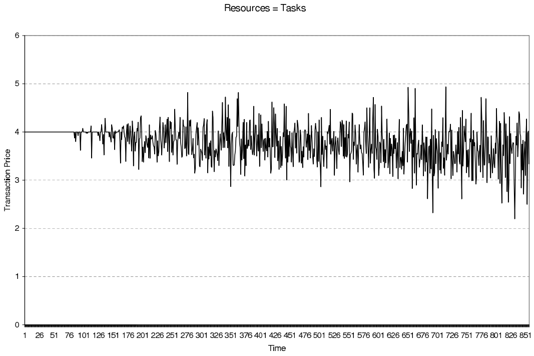

Figure 3. Balanced Network.

4.1.2 Task Intensive Network

This type of network is similar to what is called a **sellers market** which has more buyers than sellers. High prices result from this excess of demand over supply. Buyers enter into competition with each other in order to obtain scarce resources. This creates an upward pressure on the price. (see Figure 4).

29% task utilization and 99% resource utilization is the result of such task intensive network (see table 1).

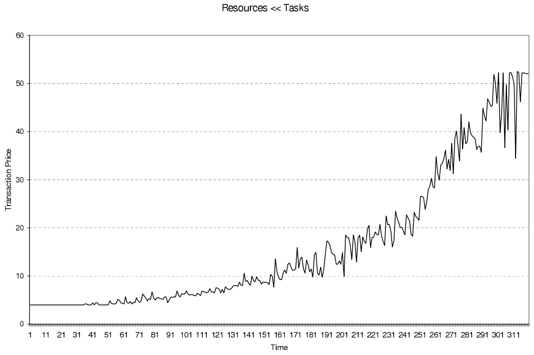

Figure 4. Task Intensive Network.

4.1.3 Resource Intensive Network

The third scenario involves a **buyer market**. In buyers market, there are more sellers than buyers and low prices result from this excess of supply over demand. Similar to the seller market, the sellers enter into competition in order to find job for their resources. A downward pressure on the price is the result, as can be observed from figure 5.

As we expected in such network, we obtained 97% task utilization whereas 22% resource utilization (see table 1).

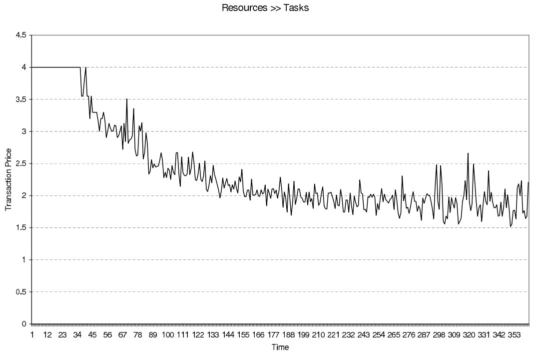

Figure 5. Resource Intensive Network.

4.2 Parameter Regime Analysis

Our model contains a number of parameters for which we have to determine a value. It is important to understand the impact of such choices on the overall system behavior. Such an exercise is called studying the parameter regime. In the pricing function (equations 2,3), u_{th} represents a threshold for resource or task utilization in each individual node. This parameter defines a critical value for task or resource utilization with which the buyers or sellers are satisfied. We run the simulation with varying threshold parameter and measured the efficiency of the entire system in each situation (see table 2). The experiments are done in a balanced network considering the values of $\alpha = \beta = 0.25$. The result shows the impact of resource or task utilization of individual nodes on the entire system utilization, as we increase the value of u_{th} the system efficiency also increases. This observation is a clear illustration of how individual behavior, namely being lazy or not, has a direct relation with the overall system efficiency.

In another experiment, we studied the pricing evolution with changing the values of α and β parameters. The impact of these technical parameters is measured in a resource intensive network with the value of $u_{th} = 0.9$ and the values of 0.75 and 0.25 for α and β (Figure 6). The α and β coefficients determine the rate at which the price changes. According to equations 2 and 3, the higher their value, the higher the Δp will be. And consequently, the higher Δp, the higher will be the price changes proposed.

Network Condition	No. of Offers	No. of Requests	No. of Matches	Resource Utilization	Task Utilization
$u_{th} = 0.9$	1033	967	856	83%	88%
$u_{th} = 0.75$	1036	964	703	67%	73%
$u_{th} = 0.25$	1006	994	271	26%	27%

Table 2. System efficiency with varying utilization thresholds in a balanced network.

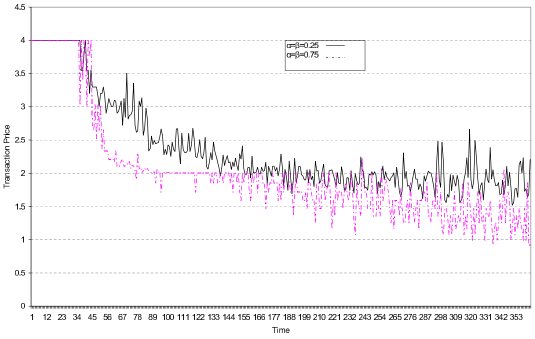

Figure 6. Price transaction with different valuse of α and β in resource intensive network.

4.3 Performance Evaluation in Comparing with a Non-price based Approach

The final evaluation we perform is to compare the market based approach to the simple First-Come First-Served mechanism. The latter is much simpler as the matchmaking is done on the basis of the time of arrival and no sorting or other manipulation is done to find the matches. Tables 1 and 3 show the resource and task efficiencies in two approaches for three conditions. The results show that the price-based mechanism is as efficient as a blind and simple matchmaking algorithm that has less constraints in finding matches. For instance in balanced network, the task and resource efficiency is between 83% and 88% for both mechanisms. The small variations are due to random generation of tasks and resources. Similar findings hold for the other network conditions.

Network Condition	No. of Offers	No. of Requests	No. of Matches	Resource Utilization	Task Utilization
Resources $\simeq$ Tasks	1000	1000	870	87%	87%
Resources $>>$ Tasks	1607	394	392	24%	99%
Resources $<<$ Tasks	349	1412	348	100%	24%

Table 3. Fist-Come, First-Served Algorithm (no price).

5 Conclusion and Future Work

In this paper, we studied a market-based resource allocation mechanism using Continuous Double Auction model. A pricing function is proposed to compute the prices. We studied the pricing behavior in balanced and unbalanced network conditions. The study of changing price in different network conditions shows that the price level reflects the state of the system and it adapts to the changing supply and demand of resources. This adaptation is achieved by increasing or decreasing the price when supply or demand are low. Besides in our model, each node can decide individually on its own utilization of resources or tasks from the system by specifying a threshold value (u_{th}). Compared to a non-market based approach, the market-based model shows a good and comparable efficiency in terms of resource and task utilization

As future work, we intend to explore different kinds of auction mechanisms, introduce heterogeneity among the agents in terms of u_{th} and

α/β. An additional extension will be the introduction of a limited budget and the computation of a transaction cost for each request. A better understanding of the different structural design choices one has for resource allocation, should enable us to provide a framework in which self management and self organization becomes possible.

References

[1] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic models for resource management and scheduling in grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1507–1542, 2002.

[2] M. D. de Assuncao and R. Buyya. An evaluation of communication demand of auction protocols in grid environments. Technical report, Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2006.

[3] D. Grosu and A. Das. Auction-based resource allocation protocols in grids. In *In Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 20–27, November 2004.

[4] G. M. Heal. Planning without prices. *Review of Economic Studies*, 36(107):347–62, 1969.

[5] U. Kant and D. Grosu. Double auction protocols for resource allocation in grids. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 366–371, 2005.

[6] J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on computers*, pages 705–717, May 1989.

[7] K. Kuwabara and T. Ishida. Equilibratory approach to distributed resource allocation: Toward coordinated balancing. In *Proceedings. Artificial Sociality; MAAMAW'92, Lecture Notes in Artificial Intelligence 830, Springer-Verlag*, pages 133–146. 1994.

[8] S. Lalis and A. Karipidis. Jaws: An open market-based framework for distributed computing over the internet. In *GRID*, pages 36–46, 2000.

[9] N. Nisan, S. London, O. Regev, and N. Camiel. Globally distributed computation over the internet - the popcorn project. In *ICDCS '98: Proceedings of the The 18th International Conference on Distributed Computing Systems*, page 592. IEEE Computer Society, 1998.

[10] E. Ogston and S. Vassiliadis. A peer-to-peer agent auction. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems Part I*, pages 151–159, July 2002.

[11] M. Preist C., Van Tol. Adaptive agents in a persistent shout double auction. In *Proc. of 1st International Conference on the Internet Computing and Economics*, pages 11–17, 1998.

[12] M. Shubik. *Game Theory in Social Sciences*. The MIT Press, Cambridge, Mass, 1983.

[13] 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.

[14] C. A. Waldspurger and W. E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Operating Systems Design and Implementation*, pages 1–11, 1994.

[15] R. Wolski, J. Brevik, J. S. Plank, and T. Bryan. Grid resource allocation and control using computational economies. In F. Berman, G. Fox, and A. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley & Sons, 2003.

[16] R. Wolski, J. Plank, J. Brevik, and T. Bryan. G-commerce: Market formulations controlling resource allocation on the computational grid. In *In Proc. International parallel and Distributed Processing Symposium (IPDPS)*, April 2001.

[17] P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.