On Economic and Computational-efficient Resource Pricing in Large Distributed Systems

Marian Mihailescu and Yong Meng Teo
Department of Computer Science, National University of Singapore
Computing 1, 13 Computing Drive, Singapore 117417
Email: [marianmi, teoym]@comp.nus.edu.sg

* This is a revised version of the paper published in the Proceedings of 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Computer Society Press, pp. 838-843, Melbourne, Australia, May 17-20, 2010.

Abstract

There is growing interest in large-scale systems where globally distributed and commoditized resources can be shared and traded, such as peer-to-peer networks, grids, and cloud computing. Users of these systems are rational and maximize their own interest when consuming and contributing shared resources, even if by doing so they affect the overall efficiency of the system. To manage rational users, resource pricing and allocation can provide the necessary incentives for users to behave such that the overall efficiency can be maximized. In this paper, we propose a dynamic pricing mechanism for the allocation of shared resources, and evaluate its performance. In contrast with several existing trading models, our scheme is designed to allocate a request with multiple resource types, such that the user does not have to aggregate different resource types manually. We formally prove the economic properties of our pricing scheme using the mechanism design framework. We perform both theoretical and simulation analysis to evaluate the economic and computational efficiency of the allocation and the scalability of the mechanism. Our simulations are validated against a prototype implementation on PlanetLab.

1. Introduction

Recent advances in computer technology and the Internet have led to the expansion of computer networks into large distributed systems. Currently, several technologies such as grid computing and cloud computing among others, are converging towards federated sharing of computing resources. In these distributed systems, resources are commodities and users can both consume and contribute with shared resources. In cloud computing, hardware and software resources are provided over the Internet on-demand, as a service, without the user having knowledge of the underlying infrastructure. Currently, cloud computing usage is increasing both in breadth, such as the number of resource types offered, and in depth, such as the number of resource providers. Thus, with an increasing number of cloud users, it is expected that more providers will offer similar services. Furthermore, with interoperability between different providers, users are able to use the same service across clouds to improve scalability and reliability. In this context, the aim of *federated clouds* is to integrate resources from different providers such that access is transparent to the user.

A fundamental problem in any federated system is the allocation of shared resources. Recent work in distributed systems acknowledges that users sharing resources are self-interested parties with their own goals and objectives [8]. Usually, these parties can exercise their partial or complete autonomy to achieve their objectives and to maximize their benefit. They can devise strategies and manipulate the system to their advantage, even if by doing so they break the rules of the system. For example, performance of peer-to-peer file sharing is affected by free-riders, users that consume more than their fair share [13]; in a computational grid, users compete for the same resources, which may result in increased waiting times [18]; some users of SETI@home, a popular distributed computing project, modified the software client to report false-negatives in order to achieve higher rankings [13].

To manage *rational users*, economics and mechanism design offer market-based approaches for pricing and allocation of shared resources [15]. Although we cannot assume rational users are trusted to follow the algorithm or protocols designed and deployed, we can assume that they participate in sharing in order to maximize their personal gain, such that incentives may be used to induce the desired behavior. Mechanism design studies how to structure incentives such that users behave according to protocols. Thus, recent work in peer-to-peer networking [7, 16], grid or cluster computing [11], resource allocation [4,18], and others, use a form of incentives to manage rational users.

Our work focuses on a dynamic pricing scheme suitable for allocating resources in federated systems, where pricing is used to manage rational users. A rational user may represent either an individual user, a group, or an organization, depending on the application context. Using the mechanism design framework, we formally prove that our pricing scheme is strategy-proof.

In many federated environments, users request more than one type of resources from different providers. In contrast to fixed pricing, where users have to manually aggregate resources from different providers, our pricing scheme is designed to allocate a request *for multiple resource types*. Moreover, in large distributed systems, resource demand and supply fluctuate as users join and leave the system. We show using simulations that in our proposed *dynamic scheme*, the user welfare, the percentage of successful requests, and the percentage of allocated resources increase when compared to fixed pricing.

We evaluate our pricing scheme both for economic and computational efficiency. In addition to theoretical analysis, we use simulations that are validated by a prototype implementation of our framework on PlanetLab. We compare our pricing scheme with combinatorial auctions, where economic efficiency is maximized at the cost of computational efficiency. In addition, we study the *scalability* of a centralized implementation and propose a distributed scheme that scales when increasing the number of users and the number of resource types.

The remainder of this paper is structured as follows. Section 2 presents related works from grid computing and distributed systems. We discuss dynamic pricing in Section 3. Our proposed framework is introduced in Section 4, while in Section 5 we evaluate the economic and computational efficiency and scalability. Finally, Section 6 contains our conclusions and future work.

2. Related Work

Resource markets have been previously proposed for sharing computational resources in the presence of rational users [6, 10, 18]. A *resource market* consists of the environment, rules and mechanisms where resources are exchanged. In this context, related works have used either bartering or pricing to exchange resources. Figure 1 presents a classification of existing economic models based on the exchange method used. In bartering, resources are exchanged directly, without using any form of currency. For example, in BOINC [3], users donate their CPU cycles by running a software client which polls a server for new jobs. In BitTorrent [7], rational users that behave selfishly and do not cooperate in sharing files are punished by other users. In contrast, in OurGrid [4], each user keeps track of other users that provide resources for their jobs, and prioritize their requests when their own resources are idle. Bartering is simple to implement and allows several types of incentives for rational users: moral incentives (volunteer computing) or coercive incentives (tit-for-tat, network of favors). However, bartering allows exchanges between a *single resource type*. For example, BitTorrent users exchange blocks from the same file, OurGrid is used for CPU cycles, etc. To exchange *multiple resource types*, our framework uses pricing and common currency to express the value of each resource type.

Pricing is the process of computing the exchange value of resources relative to a common form of currency. Economic models for the allocation of shared resources may use fixed or dynamic pricing. When using fixed pricing, each resource type has a predefined price, set by the seller. For example, Amazon provides disk space for $0.15/GB. In contrast, when using dynamic pricing, the resource price is computed for each request according to the pricing mechanism used. More specifically, a resource type can have the same price

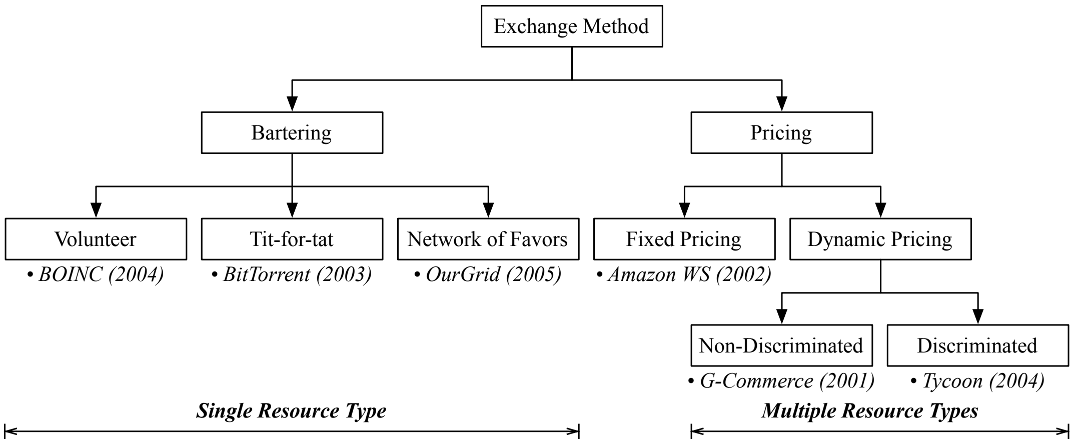

Figure 1: Economic-inspired Resource Allocation Models

for all resource providers (non-discriminated pricing), or payment is computed differently for each resource provider (discriminated pricing). Pricing schemes use financial incentives in addition to payments to motivate rational users to be truthful.

Several market-based allocation systems for grids, such as Nimrod/G [5], use bargaining or negotiation to determine the resource price. The advantage of this approach is that sellers and buyers communicate directly, without a third party mediating an allocation. The seller attempts to maximize the resource price, while the buyer strives to minimize it. However, communication constitutes the main disadvantage of bargaining: in a large dynamic market, each buyer has to negotiate with all sellers of a resource type in order to maximize his utility. The communication costs grow further when a buyer requires more than one resource types. Thus, scalability becomes a major issue when increasing the number of users or the number of resource types in a request. To address scalability, we propose a distributed reverse-auction mechanism built on top of a peer-to-peer overlay, such that any user can become a market-maker and auctions for different resource types can take place simultaneously.

In contrast to resource sharing systems used in research and academic communities or for personal benefit, cloud computing has been put into commercial use and its economic model is based on fixed pricing. Previous unsuccessful cloud computing attempts, such as Intel Computing Services, required users to negotiate written contract and pricing. However, current online banking and currency transfer technologies allow cloud providers to use fixed pricing, with buyer payments made online using a credit-card. Federated clouds can be formed by combining private clouds to provide users with resizeable and elastic capacities. Currently, companies such as Amazon operate as standalone clouds service providers. However, in a federated cloud, any globally distributed user can both offer and use cloud services. A user is either an individual, a group, or an organization, depending on the application context. In contrast with fixed pricing, our scheme is designed to reflect the dynamic value of shared resources when supply and demand fluctuates.

3. Dynamic Pricing Mechanisms

Market-based resource allocation mechanisms based on pricing introduce several economic and computational challenges. From a computational perspective, a mechanism must compute in polynomial time the allocation of multiple resource types while maximizing the number of allocated resources and satisfied requests. However, an optimal allocation mechanism for multiple resource types such as combinatorial auctions requires a NP-complete algorithm [14]. Accordingly, many systems share only one resource type, such as CPU cycles in volunteer computing, and file blocks in file-sharing networks.

From an economic perspective, the desirable properties for resource allocation are: *individual rationality*, *incentive compatibility*, *budget balance* and *Pareto efficiency* [12]. In an individual rational allocation mechanism, rational participants gain higher utility by participating in resource sharing than from avoiding it. Incentive compatibility ensures that the dominant strategy for each participant is truth-telling. Budget-balance verifies that the sum of all payments made by buyers equal the total payments received by the sellers. *Pareto efficiency*, the highest economic efficiency, is achieved when, given an allocation, no improvement can be made that makes at least one participant better off, without making any other participant worse off. However, according to the Myerson-Satterthwaithe impossibility theorem [12], no mechanism can achieve all four properties together. Accordingly, related works have traded incentive compatibility [6], economic efficiency [10] or budget-balance [14]. In contrast, our approach is to focus on achieving individual rationality, incentive compatibility and budget balance using a computationally efficient algorithm that can allocate buyer requests for multiple resource types.

In a federated system market, *dynamic pricing* sets resource payments according to the forces of demand and supply. Moreover, the use of dynamic pricing facilitates sellers to provide multiple resource types. Early cloud services such as Sun Grid Compute Utility were restricted to one resource type, e.g. CPU time. More recent services, such as Amazon S3 and EC2, introduced more resource types, i.e. storage and bandwidth. Currently, Amazon has expanded its offer to 10 different virtual machine instance configurations, with different prices for each configuration, and practice tiered pricing for storage and bandwidth. We see this as the first step towards dynamic pricing, where users can request for custom configurations with multiple resource types.

The resource market in federated environments consists of many *resource types*. Figure 2 shows a simplified resource model where a buyer request can consist of many resource types and many resource items for each type. A resource type is loosely defined, and can be a hardware resource, a software service, or a combination. We consider the example of a New York Times employee that used 100 EC2 instances to convert 4 TB of TIFF files to the PDF format. To complete his job, the user required multiple resource types

Resource_Type = **Description**
Publish = **Seller_Address**, **Resource_Type**, **Items**, **Cost**
Request = **Buyer_Address**, (**Resource_Type**, **Items**)+, **Price**

Figure 2: Simplified Model for Multiple Resource Types Buyer Request

(storage from Amazon S3 and computational power from Amazon EC2), and multiple items (100 Amazon EC2 instances) to complete his task. In this example, we assume Amazon EC2 provides ten resource types, called *instances*. A *small instance* consists of 1 EC2 compute unit (approx. 1 GHz CPU), 1.7 GB memory and 160 GB storage, and is priced at \$0.10/hour, while an *extra-large instance* consists of 8 EC2 compute units, 15 GB memory and 1.6 TB storage, and is priced at \$1.00/hour.

4. Strategy-proof Auction Framework

In the context of federated sharing systems, we propose a strategy-proof dynamic pricing mechanism for allocating shared resources with multiple resource types. We assume a resource market where rational users can both provide (sellers) and utilize resources (buyers). Rational users represent either an individual or an organization. Interoperability provides the buyers with uniformity and elasticity. Thus, a buyer request for a large number of resources can be met by more than one seller.

In [17], we define a mechanism design problem that describes a resource sharing system where rational users can be both buyers and sellers of resources. Given a set of alternative choices, a rational user selects the alternative that maximizes the expected value of his utility function. In our mechanism design problem, the utility functions are determined by the seller costs and the buyer budget, respectively.

Definition. (The Market-based Resource Allocation Problem). *Given a market containing requests submitted by buyers and resources offered by sellers, each participant is modeled as a rational user* i *with private information* t_i. *A seller has private information* t_s^r, *the underlying costs for the available resource* r, *such as power consumption, bandwidth costs, etc. The buyer's private information is* t_b^R, *the maximum price the buyer is willing to pay such that resources are allocated to satisfy its request* R. *Seller's* i *valuation is* t_i^r *if the resource* r *is allocated, and* 0 *if not. Similarly, buyer's* i *valuation is* t_i^R *of the request* R *is allocated, and* 0 *if not. For a particular request* R, *the goal is to allocate resources such that the underlying costs are minimized.*

Using the mechanism design framework in which the market-based resource allocation problem is formulated, we propose a reverse auction-based mechanism, which we prove formally to be individual rational, incentive compatible and budget-balanced. A mechanism that is both individual rational and incentive compatible is known as *strategy-proof.*

Auctions are usually carried out by a third party, called the *auctioneer* or the *market-maker*, which collects the bids, selects the winners and computes the payments. Our approach to model the market-maker consists of two parts. Firstly, we focus on the economical and computational advantages of dynamic pricing, and, for simplicity, we consider a centralized market-maker to which sellers publish resources and buyers send requests. Then, to address the scalability problem of a centralized system, we introduce distributed auctions, where more than one market-makers are able to auction at the same time. Given that buyers and sellers are globally distributed, we adopt a peer-to-peer approach, where resource discovery is facilitated by an overlay network.

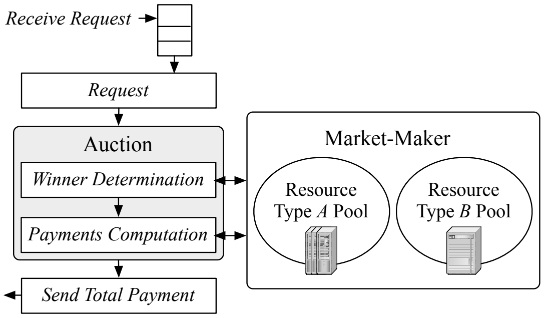

Figure 3: Two-step Auction Protocol

As shown in Figure 3, the reverse auction contains two steps: *winner determination* and *payment computation*. Winner determination decides which sellers are selected for allocation, based on the published price, such that the underlying resource costs are minimized. However, to achieve strategy proof using financial incentives, the actual payments for the winning sellers are determined in the second step, based on the market supply for each resource type.

Accordingly, the payment for a seller is determined for each resource type using a VCG-based [9] function, which verifies the incentive-compatibility property for sellers:

$$p_s = \begin{cases} 0, & \text{if seller } s \text{ does not contribute with} \\ & \text{resources to satisfy the request} \\ c_{M|s=\infty} - c_{M|s=0} & \\ & \text{if seller } s \text{ contributes with} \\ & \text{resources to satisfy the request} \end{cases}$$

where:
$c_{M|s=\infty}$ is the lowest cost to satisfy the request without the resources from seller s;
$c_{M|s=0}$ is the lowest cost to satisfy the request when the cost of resources from seller s resources is 0.

To achieve the incentive-compatibility property for buyers, we select the requests using the first-come-first-serve strategy [17]. To obtain budget-balance, the buyer payment function is the sum of all seller payments:

$$p_b = -\sum_{s \in S} p_s$$

where S is the set of winner sellers.

5. Simulation Evaluation

We evaluate the proposed pricing mechanism both for economic and computational efficiency. Firstly, we compare our pricing scheme with combinatorial auctions, using jCase [1], a java combinatorial auction simulator. Next, we implement the proposed framework on top on FreePastry [2], an open-source DHT overlay network environment.

FreePastry allows us to analyze the performance of the proposed pricing scheme in two different environments: the Free Pastry simulator, where we can simulate a large resource market, and PlanetLab, where we deploy our framework to validate the simulator results on a smaller number of distributed nodes. Using simulation, we can compare our dynamic pricing scheme with fixed pricing, currently used by many cloud providers.

For simplicity, we use a centralized market-maker to compare the economic and computational advantages of dynamic pricing. A centralized implementation has the advantage of allowing the measurement of economic and computational efficiency with a simple setup for a large simulated network. Moreover, the use of a peer-to-peer underlying overlay such as FreePastry allows us to address the scalability issue in our future work. Publish and request messages are sent to the market-maker node using the FreePastry routing process, which then performs the reverse auctions using the first-come-first-serve policy and computes the payments using the functions in Equation 4 and Equation 4, respectively.

5.1. Economic and Computational Efficiency

Economic efficiency is a global measure and represents the *total welfare* for both buyers and sellers. More specifically, there are two factors that affect the economic efficiency: *i)* average user welfare; and *ii)* number of successful requests ($\%succ$), for buyers, and number of allocated resources ($\%alloc$), for sellers.

Computational efficiency is a major design criteria in the allocation of shared resources. Optimal mechanisms such as combinatorial auctions [14] are not feasible since the winner determination algorithm is NP-complete. On the other hand, fixed pricing has the advantage of eliminating the time incurred by payment computation. To determine the time cost of the algorithm used by the proposed mechanism, we perform a theoretical analysis of the runtime complexity for the winner determination and the buyer and seller payment functions, and validate the theoretical analysis with experimental results.

Pricing Mechanism	**Users**	**Total Welfare**	**Runtime**	**%succ**	**%alloc**
combinatorial	20	2470	9.6m	44.5	44.8
	40	6321	2.5h	52.5	57.2
	80	14384	67.4h	54.2	64.0
proposed	20	1871	1s	36.3	32.5
	40	5438	3s	48.5	55.3
	80	11561	5s	52.8	68.0

Table 1: Comparison with Combinatorial Auctions

Table 1 shows a comparison between the efficiency of combinatorial auctions and the proposed pricing mechanism using jCase. Our results show that the proposed scheme is close to the optimal economic efficiency, while achieving a feasible runtime.

5.2. Dynamic Pricing

To evaluate the impact of dynamic pricing, we compare the percentage of successful buyer requests, the percentage of allocated seller resources and the average buyer welfare with fixed pricing. Our results presented in Table 2 show that in different market conditions, dynamic pricing obtains better allocation results which result in increased economic efficiency.

Metric	**Market Condition**	**Pricing Scheme**	
		Fixed	**Dynamic**
% succ. requests	Under-demand	47.4	86.3
	Balanced	47.1	62.9
	Over-demand	47.4	41.5
% alloc. resources	Under-demand	23.4	44.1
	Balanced	46.5	62.5
	Over-demand	93.1	74.2
avg. buyer welfare	Under-demand	4.7	9.4
	Balanced	4.7	6.3
	Over-demand	4.7	4.8

Table 2: Comparison with Fixed Pricing

5.3. Scalability

The proposed pricing mechanism can allocate resources for buyer requests containing more than one resource types (multiple resource type allocations), while achieving incentive-compatibility and budget-balance. However, one of the limitations of the auction model is the requirement for a centralized market-maker, which prevents scalability. We refer to *vertical* *scalability* when increasing the number of resource types, and *horizontal scalability* when increasing the number of users. We propose a distributed auction model built on a DHT overlay network, where any peer can play the role of a market maker. In this context, we analyze the vertical and horizontal scalability both theoretical and experimental.

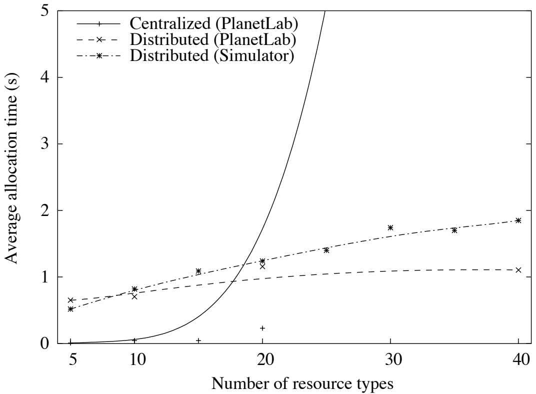

Figure 4: Scalability of the Distributed Scheme

Our simulation results show that the distributed implementation is feasible in large-scale systems. Figure 4 shows the vertical scalability of the distributed scheme by comparing the average allocation time with the centralized implementation.

6. Conclusions and Future Work

In this paper we present a strategy-proof resource allocation mechanism for pricing and allocation of shared resources. To address the needs of dynamic large systems, we propose a distributed market framework built on top of a peer-to-peer overlay. Our pricing mechanism for resource allocations of requests with multiple resource types: *i)* provides incentives for rational users; *ii)* achieves budget-balance; *iii)* is computationally efficient and *iv)* is vertically and horizontally scalable. In summary, the main contributions of our work are:

- *Efficient allocation of multiple resource types.* Optimal allocation algorithms are often infeasible to implement, and resource providers have chosen either to use fixed pricing and negociations, or to create separate markets for different resource types. We propose a computational efficient pricing mechanism and study the benefits of using dynamic pricing in a large market where a user request contains multiple resource types.
- *Formal approach to market-based resource allocation.* We express the problem of allocating shared resources in federated systems as a mechanism design optimization problem and formally analyze the economic properties achieved by our resource allocation mechanism.
- *A scalable distributed market framework.* Centralized mechanisms have proved impractical in large distributed systems. Our framework built on top of a peer-to-peer overlay efficiently divides resource information in a distributed market. This allows vertical scalability, i.e. when increasing the number of resource types in a request, and horizontal scalability, i.e. when increasing the number of users in the system.

For future work, we focus on: further analysis of the distributed pricing mechanism efficiency under different workload conditions including high request churn rates; a detailed evaluation of the economic efficiency trade-off of the proposed mechanism; and the study the strategic properties of the distributed resource location mechanism.

References

[1] jCase: Java Combinatorial Auction Simulation Environment, http://www.schnizler.com/jcase.html, 2009.

[2] Pastry: a scalable, decentralized, self-organizing and fault-tolerant substrate for P2P applications, http://www.freepastry.org, 2009.

[3] D.P. Anderson. BOINC: A System for Public-Resource Computing and Storage. In *5th IEEE/ACM Intl. Workshop on Grid Computing*, pp. 4–10, Pittsburgh, USA, 2004.

[4] N. Andrade, W. Cirne, F. V. Brasileiro, and P. Roisenberg. OurGrid: An Approach to Easily Assemble Grids with Equitable Resource Sharing. In *Proc. of the 9th Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 61–86, Seattle, USA, 2003.

[5] 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 Intl. Conf. on High Performance Computing in Asia-Pacific Region*, pp. 283–289, Beijing, China, 2000.

[6] B. N. Chun and D. E. Culler. Market-based Proportional Resource Sharing for Clusters. Technical Report UCB/CSD-00-1092, EECS Department, University of California, Berkeley, USA, 2000.

[7] B. Cohen. Incentives Build Robustness in BitTorrent. In *Proc. of the 1st Workshop on Economics of P2P Systems*, Berkeley, USA, 2003.

[8] A. R. Dani, A. K. Pujari, and V. P. Gulati. Strategy Proof Electronic Markets. In *Proc. of the 9th Intl. Conference on Electronic Commerce*, pp. 45–54, Minneapolis, USA, 2007.

[9] E. Elkind. True Costs of Cheap Labor Are Hard to Measure: Edge Deletion and VCG Payments in Graphs. In *Proc. of the 7th ACM Conf. on Electronic Commerce*, pp. 108–116, Vancouver, Canada, 2005.

[10] K. Lai, B. A. Huberman, and L. R. Fine. Tycoon: A Distributed Market-based Resource Allocation System. Technical Report cs.DC/0404013, HP Labs, Palo Alto, USA, 2004.

[11] L. Lin, Y. Zhang, and J. Huai. Sustaining Incentive in Grid Resource Allocation: A Reinforcement Learning Approach. In *Proc. of the IEEE Intl. Symposium on Cluster Computing and the Grid*, pp. 145–154, Rio de Janeiro, Brazil, 2007.

[12] R. B. Myerson and M. A. Satterthwaite. Efficient Mechanisms for Bilateral Trading. *Journal of Economic Theory*, 29(2):265–281, 1983.

[13] S. J. Nielson and S. A. Crosby. A Taxonomy of Rational Attacks. In *Proc. of the 4th Intl. Workshop on Peer-to-Peer Systems*, pp. 36–46, Ithaca, USA, 2005.

[14] N. Nisan. Bidding and Allocation in Combinatorial Auctions. In *Proc. of the 2nd ACM Conference on Electronic Commerce*, pp. 1–12, Minneapolis, USA, 2000.

[15] N. Nisan and A. Ronen. Algorithmic Mechanism Design (extended abstract). In *Proc. of the 31st Annual ACM Symposium on Theory of Computing*, pp. 129–140, Atlanta, USA, 1999.

[16] J. Shneidman and D. C. Parkes. Rationality and Self-Interest in Peer to Peer Networks. In *Proc. of the 2nd Intl. Workshop on Peer-to-Peer Systems (IPTPS'03)*, pp. 139–148, Berkely, USA, 2003.

[17] Y. M. Teo, and M. Mihailescu. A Strategy-proof Pricing Scheme for Multiple Resource Type Allocations. In *Proc. of the 38th Intl. Conference on Parallel Processing*, pp. 172-179, Vienna, Austria, 2009.

[18] 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 Intel. Parallel and Distributed Processing Symposium*, pp. 46–54, San Francisco, USA, 2001.