Combinatorial Auction-Based Protocols for Resource Allocation in Grids

Anubhav Das
Dept. of Computer Science
Wayne State University
5143 Cass Avenue, Detroit, MI 48202.
anubhav@cs.wayne.edu

Daniel Grosu
Dept. of Computer Science
Wayne State University
5143 Cass Avenue, Detroit, MI 48202
dgrosu@cs.wayne.edu

Abstract

In this paper, we introduce the combinatorial auction model for resource management in grids. We propose a combinatorial auction-based resource allocation protocol in which a user bids a price value for each of the possible combinations of resources required for its tasks execution. The protocol involves an approximation algorithm for solving the combinatorial auction problem. We implement the new protocol in a simulated environment and study its economic efficiency and its effect on the system performance.

1. Introduction

Grid systems are defined as next generation computing platforms for solving large scale problems in science and engineering. Grids are based on the concept of flexible, secure, and coordinated resource sharing among dynamic collections of institutions distributed across the world known as virtual organizations [9]. The participating resources in a Grid may be computational resources, data storage or computer networks. It is difficult to design optimal grid resource allocation mechanisms which meet the objectives of both users and resource owners. The reason being that users and resource owners have different requirements and objectives. Several economic-based resource allocation mechanisms that address this complex problem have been proposed [3, 14, 20]. The trading and brokering policies on which these mechanisms rely help to match the different requirements of end users and resource owners. They are better than the classical resource management schemes because they are decentralized in structure and they use incentives for resource owners to contribute resources.

The two main economic models used in the context of resource management in a distributed system are commodities markets and auctions. In the *commodities market model* described in [3], for each unit of resource consumed by a user, a publicly agreed price is charged to the user. In case of the *auction model* the participating resource owners (service providers) and users agree privately on the selling price. Compared to other approaches for resource allocation, auctions have many advantages, they are decentralized in nature, they require little global information and are easy to implement. The previous studies on auction-based resource allocation protocols [12, 13, 19] have considered only two types of auctions, one-sided auctions (e.g. First Price and Vickrey auctions) and double-sided auction (e.g. Double auction) and compared them with other economic-based and conventional models. To improve the economic efficiency and maximize the revenue in a Grid, instead of a user bidding for each task separately (as in the case of one-sided auction and double-sided auction), the user can bid a price value for each of the possible combinations of resources on which its tasks can be executed. This is the scenario in a combinatorial auction where the winner determination algorithm finds an optimal allocation of resources for the tasks, increasing the economic efficiency and maximizing the revenue. Therefore, in this paper we investigate a combinatorial auction-based resource allocation protocol in terms of its suitability in grid systems, economic efficiency and system performance. We define the combinatorial auction model in which the main participants are resource owners, users and auctioneers. The *resource owners* provide grid services like computational power, data storage, software or computer networks. The *users* have applications for which they require the services provided by resource owners. In the proposed resource allocation protocol each user can bid for each possible combination of resources required for its tasks execution. Each user has a *broker* who manages and schedules user's jobs in the Grid, generates each possible combination of resources for the users tasks and the price value for each combination that the user agrees to pay in the auction and hands payments to resource providers. The responsibility of an auctioneer includes setting the rules of the auction and conducting the combinatorial auction. The auctioneer first collects bids from brokers participating in the auction and then decides the best allocation, using an approximation algorithm for solving the winner determination problem. Finally it collects the payments from the users which won the auction. Besides it also interacts with the local scheduler to schedule the jobs of the user who wins in the auction.

Using the proposed combinatorial auction model we design a combinatorial auction based resource allocation protocol. To evaluate this resource allocation protocol, the most simple and reliable way would be to perform real experimentation. This involves scheduling and executing real applications on real resources. The problem with this approach is that, firstly, real applications may run for long time which is time consuming, secondly, we cannot explore a wide range of different resources by the means of experimentation on real resources. Finally, due to the varying nature of load on resources the results obtained will not be repeatable. Thus the most viable approach is to resort to simulation. In order to perform simulations we developed a simulator based on the SimGrid simulation framework [5]. The simulator allows us to evaluate the combinatorial auction-based resource allocation protocol in terms of economic efficiency and system performance.

Related work. Economic-based resource management systems have been investigated by several researchers in [1, 6, 7, 8, 11, 12, 17, 19]. A comprehensive survey of economic models for resource management in distributed systems can be found in [3]. Wolski *et al.* [19] investigated the problem of resource allocation in grids under two economic models: commodities markets and auctions. They compared these two models in terms of price stability and market equilibrium. A detailed survey on combinatorial auctions and related issues can be found in [18]. Several methods to solve a combinatorial auction problem have been proposed by Fujishima *et al.* [10], Sandholm *et al.* [16], Rothkopf *et al.* [15] and Andersson *et al.* [2]. The first authors use dynamic programming to solve the winner determination problem. The next two approaches use refinements by pruning the search tree and introducing additional bounding heuristics. The last approach uses integer programming to solve the combinatorial auction problem. An efficient approximate allocation algorithm for combinatorial auction is presented in Nisan *et al.* [21]. Two simulation toolkits GridSim [4] and SimGrid [5] provide core functionalities to build simulators for studying resource allocation protocols in Grid environments.

Our contributions. The previous work on auction-based protocols for resource allocation focused only on the comparison of two types of auctions (one-sided auctions and double-sided auctions) with other economic models. This paper studies a different class of auctions called combinatorial auctions in the context of resource management in grids. In a combinatorial auction-based protocol a user bids one value for a set of resources required to complete a job composed of several tasks instead of bidding on individual resources for each task. The motivation behind investigating combinatorial auctions is that the economic efficiency is enhanced compared to other auction models and also the revenue is maximized. In this paper we present the combinatorial auction model and a combinatorial auction-based resource allocation protocol (based on the proposed model). We simulate the protocol using the SimGrid simulation framework and we evaluate it in terms of economic efficiency and system performance.

Organization. The paper is structured as follows. Section 2 presents the combinatorial auction allocation model and the combinatorial auction-based resource allocation protocol. In Section 3 we give a brief description of the SimGrid simulation environment. Section 4 presents the simulation of the proposed combinatorial auction-based resource allocation protocol using the SimGrid simulator and the experimental results. In Section 5 we draw conclusions and present future research directions.

2. Combinatorial Auction Based Resource Allocation: Model and Protocol

2.1 Combinatorial Auction Allocation Model

The main participants in the auction model (Figure 1) are: Grid Service Providers (GSP), User Brokers (UB) and Local Markets for Auctions (LMA). In the following we present each of these participants and describe their role in the model and their characteristics.

User Broker (UB): Each grid user has a User Broker. The User Broker is responsible for resource discovery, generating all possible combinations of resources for the user tasks according to their requirements, generating the corresponding bid value for each combination, submitting the bid and the corresponding combination to an external auctioneer, sending user jobs to resources, collecting the results and providing the user with a uniform view of grid resources. There are three components of the user broker:

Job Management Agent: It is responsible for user interaction, job creation, submission and monitorization. It also coordinates the mechanism analysis and selection, resource discovery and the bidding process. When the jobs complete it collects the results of the computation.

Resource Discovery Agent: It is responsible for resource discovery. It sends a request to the Local Market for Auctions for each of the users' tasks. The Local Market for Auctions sends back the list of resources that match the requirements of the task.

Auction Analysis and Selection Agent: It is responsible for analyzing the auction information submitted by the Local Market for Auctions. Based on the user requirements and on the properties of the auctions it selects a combinatorial auction (run by an External Auctioneer) in which the user will participate.

Bidding Agent: It is responsible for generating all possible combinations of resources based on the list of resources returned by the LMA for each of the user tasks. For each combination of resources it generates a bid value which is within the user budget. It sends the corresponding combination and the bid value to the External Auctioneer.

In our model we assume that there are m users, $U_0, U_1, \ldots, U_{m-1}$, each having a certain number of jobs ready to be submitted for execution. The jobs of an user U_j are composed of tasks denoted by $T_{jl}, l = 0, 1, \ldots, m_j - 1$, where m_j is the total number of tasks generated by U_j. The tasks of U_j are characterized by three parameters:

(i) *Execution time* (t_{jl}): It is defined as the execution time of task T_{jl} (in seconds) on a reference resource R_{ref}. The slowest resource in the system is considered as the reference resource.

(ii) *Group preference* (r_{jl}): It is defined as the index of the group of resources which meet the requirements of task T_{jl}. A group consists of all resources that have the same characteristics. If task T_{jl} needs to be executed on a resource from a group G_g of resources then $r_{jl} = g$.

(iii) *Task budget* (q_{jl}): It is defined as the maximum amount user U_j can pay to any resource for executing T_{jl}. It is given in 'grid dollars' (G$).

User U_j has a total budget Q_j in 'grid dollars' which is given by $Q_j = \sum_{l=0}^{m_j-1} q_{jl}$. U_j's bid b_c for the tasks on a combination of resources is within the task budget, i.e. $b_c \leq Q_j$. If any of the possible combinations of resources for which user U_j bids b_c is in the allocation determined by EA then the user pays b_c, otherwise it pays nothing.

Grid Service Providers (GSP): GSPs contribute their resources to the Grid and charge the users for services. The Auctioneer Agent is responsible for posting the GSP's characteristics on the LMA. Also it receives the result of the auction from EA and accepts jobs from the users who won the auction. Another role of the Auctioneer Agent is to coordinate the admission control and resource scheduling. We consider that GSP$_i$ is responsible for resource R_i's management ($i = 0, 1, \ldots, n-1$). Each resource R_i is characterized by the following:

(i) *Processing rate* (s_i): It is given in MIPS.

(ii) *Reservation price* (p_i): It is defined as the minimum price accepted by GSP$_i$ for one second of job execution. We consider here the following pricing strategy: the higher the processing rate the higher the reservation price.

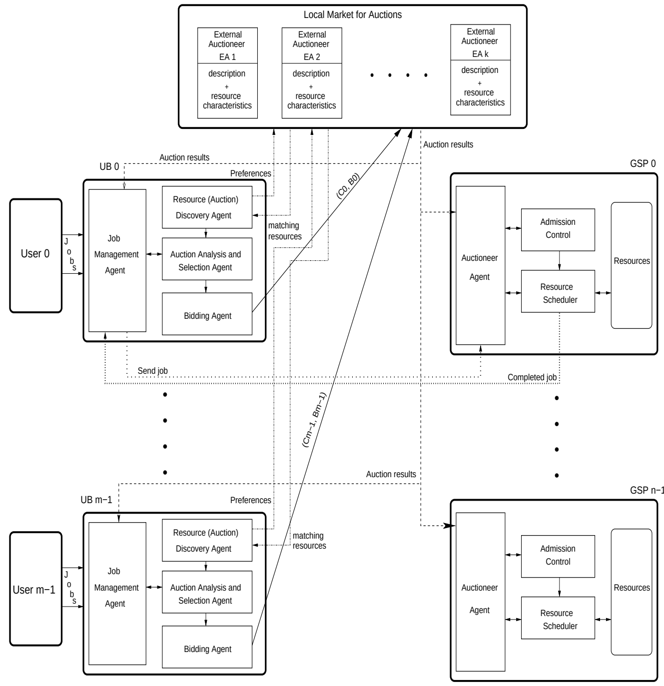

Figure 1. Combinatorial auction allocation model

(iii) *Cost* (C_{ijl}): Represents the cost incurred by GSP$_i$ when executing task T_{jl}. It is defined as: $C_{ijl} = p_i \frac{s_{ref}}{s_i} t_{jl}$.

(iv) *Resource Profit* (P_{ijl}): It is the profit gained by GSP$_i$ by executing task T_{jl}. It is defined as $P_{ijl} = PY_{ijl} - C_{ijl}$, where PY_{ijl} is the payment given to GSP$_i$ by user U_j for executing T_{jl}. The payment is given in G$. The total profit for a resource is the sum of all the profits obtained by executing all the assigned tasks.

Local Market for Auctions (LMA): It provides support for GSPs to post their characteristics, and enables the users to find the right resources that match their requirements. LMA takes a request for a task from a user broker specified in an appropriate language and returns the list of resources that match the requirements of the task. LMA consists of several *External Auctioneers* (EA). An External Auctioneer is responsible for collecting the combination of resources and the corresponding price value from the user brokers. Based on the corresponding price value of the bundles (combination of resources) it runs the algorithm for winner determination for the combinatorial auction [21]. It then sends the results to the user brokers who have won the auction and informs the GSPs which are going to receive tasks for execution.

2.2 Combinatorial Auction Based Resource Allocation Protocol

The Grid system consists of a set $\mathcal{R}$ of n resources $\mathcal{R}$ =$\{R_0,\ R_1,\ \ldots,\ R_{n-1}\}$ where each R_i belongs to a group depending on its characteristics, such as speed, memory, architecture, etc. All resources belonging to the same group have the same characteristics. There are t groups of resources $G_0, G_1, \ldots, G_{t-1}$. We consider m users $\{U_0, U_1, \ldots, U_{m-1}\}$ and their associated user brokers $\mathcal{UB} = \{UB_0, UB_1, \ldots, UB_{m-1}\}$, where each user has some application job J_j composed of the following tasks J_j=$\{T_{j1}, T_{j2}, \ldots, T_{jl}\}$, where $l = |J_j|$. Each T_{jl} has a group preference $r_{jl} = g$, where g is the index of the group. The combinatorial auction-based resource allocation protocol consists of four phases as follows:

CAP Protocol

Phase I: Request information from LMA.

1. $UB_j, j = 0, 1, \ldots, m-1$ does the following:

   For each task $T_{jl} \in J_j$, sends a message to LMA containing the requirements r_{jl} of T_{jl}. This message requests from LMA a list of resources that match the requirements for each task $T_{jl} \in J_j$.

2. For each requirement $r_{jl} = g$ of task T_{jl} received from user broker UB_j, EA sends the list of resources from the group G_g=$\{R_{g0},\ R_{g1},\ \ldots,\ R_{gk}\}$, $k = |G_g|$, that matches the preferences of T_{jl}. Thus UB_j receives one such list for every task.

Phase II: Generate bundles.

1. $UB_j, j = 0, 1, \ldots, m-1$ does the following:
   (a) Generate all possible combinations of resources (bundles), c_{jx} = $\{R(T_{j1}),\ R(T_{j2}),\ \ldots,\ R(T_{jl})\}$, where x is the combination index and R is a mapping which maps a task to a resource from the group G_g for which $g = r_{jl}$. The set of all such combinations is denoted by C_j.
   (b) For each combination of resources $c_{jx} \in C_j$ generate a price value $b_j(c_{jx})$ within the U_j's budget. The set of all such bids is denoted by B_j.
   (c) Send C_j and B_j to EA.

Phase III: Determine the Allocation.

1. EA receives B_j and C_j from each user broker UB_j, $j = 0, 1, \ldots, m-1$.
2. Using the information received, EA creates the following instance of the combinatorial auction problem (CAP):

$$\max \sum_{j \in \mathcal{UB}} \sum_{S \in C_j} b_j(S) y(S, j) \qquad (1)$$

   such that

$$\sum_{S \ni i} \sum_{j \in \mathcal{UB}} y(S, j) \leq 1 \qquad \forall i \in \mathcal{R} \qquad (2)$$

$$\sum_{S \in C_j} y(S, j) \leq 1 \qquad \forall j \in \mathcal{UB} \qquad (3)$$

$$y(S, j) = \{0, 1\} \qquad \forall S \in C_j, \quad j \in \mathcal{UB} \quad (4)$$

   where (2), (3) and (4) are the constraints that needs to be satisfied by the solution. The first constraint ensures that a resource is not allocated to more than one user, while the second constraint ensures that the same combination of resources is not allocated to more than one user.

   The variable $b_j(S)$ is the price value that UB_j bids for the combination S where $b_j(S) \in B_j$ and $S \in C_j$. The variable $y(S, j)$ = 1 if the bundle S $\in C_j$ is allocated to the user U_j in the solution, and $y(S, j)$ = 0 otherwise.
3. Given the above (CAP) problem EA determines the allocation using the approximation algorithm proposed in [21]. The output of this algorithm is a collection of pairwise disjoint bundles denoted by W =$\{w_1,\ w_2,\ \ldots,\ w_k\}$ where k=$|W|$. Each w_j is a bundle (set of resources) which would be allocated to the user that requested it.
4. EA sends the results of the auction to the users. It sends the bundle w_j to the user broker that requested the bundle w_j and it sends reject messages to all the other users.

Phase IV: Allocation of tasks to the winning resources.

1. $UB_j, j = 0, 1, \ldots, m-1$ does the following:
   (a) If UB_j receives a bundle w_j it sends the jobs to the GSPs present in the bundle w_j.
   (b) UB_j sends the payment for bundle w_j to EA which distributes it fairly among the GSP's present in the bundle w_j.

To solve the combinatorial auction problem, we used the approximation algorithm described in [21]. The algorithm finds an allocation close to the optimal allocation and works in two phases. In the first phase it determines an approximation on the linear programming relaxation of the combinatorial auction. In the last phase, called *hill climbing phase* the algorithm improves the solution quality by running a sequence of greedy procedures on the allocation obtained in the first phase.

3. Simulation Environment

To investigate the combinatorial auction based allocation mechanism presented in this paper we use the SimGrid simulation framework proposed by Casanova in [5]. The SimGrid toolkit provides tools for developing and evaluating resource allocation algorithms in heterogeneous distributed environments. It facilitates the creation of realistic resource models with different resource configurations where each resource can have varying performance characteristics like workload, data storage capacity or network bandwidth.

In the following we describe the SimGrid implementation of our simulator. The participants in the combinatorial auction based model i.e, resource owners, users and the auctioneer are represented as pthread entities. The resource owners are characterized by processing rates and reservation prices and are created in the system using the functionalities provided for hosts in the SimGrid. The users are characterized by the computational tasks which are to be executed. The computational tasks characterized by execution time, budget and group preference are created using the functionalities provided for tasks in the SimGrid. Based on the bids received from the active users for all the possible combinations of resources for their tasks, the auctioneer runs the approximate algorithm to find an allocation solution close to the optimal one. The auctioneer conducts multiple rounds of auction until all the tasks in the system are scheduled. Once all the tasks in the system have been scheduled, they are executed using the simulation functionality in SimGrid.

4. Experimental Results

In this section we investigate by simulation the combinatorial auction based resource allocation protocol using the SimGrid simulator. The simulated grid environment consists of 15 resources/GSPs $\{R_0, R_1, \dots, R_{14}\}$, divided into five groups $\{G_0, G_1, G_2, G_3, G_4\}$. Each group is characterized by a different processing rates and reservation prices as given in Table 1. The processing rates are within the range [500, 2500] MIPS which characterizes a real grid environment. The resources with higher processing speeds have higher reservation prices compared to resources with low processing speeds because they can execute more portion of a job in one second thus incurring more cost to them.

Group	G_0	G_1	G_2	G_3	G_4
s_i (MIPS)	500	1000	1500	2000	2500
p_i (G$/sec)	5	10	15	20	25
Resources	R_0	R_1	R_2	R_3	R_4
	R_5	R_6	R_7	R_8	R_9
	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}

Table 1. Resources in the system.

Users	Tasks	Budget	Total Budget
U_0	T_0	468	1012
	T_1	117	
	T_2	427	
U_1	T_3	258	728
	T_4	51	
	T_5	419	
U_2	T_6	317	826
	T_7	414	
	T_8	95	
U_3	T_9	107	419
	T_{10}	312	
U_4	T_{11}	166	377
	T_{12}	211	
U_5	T_{13}	457	834
	T_{14}	377	
U_6	T_{15}	242	242
U_7	T_{16}	306	1135
	T_{17}	174	
	T_{18}	365	
	T_{19}	290	
U_8	T_{20}	364	986
	T_{21}	316	
	T_{22}	231	
	T_{23}	75	
U_9	T_{24}	155	626

Table 2. Users and tasks in the system.

4.1 User Payments

We consider ten users $\{U_0, U_1, \dots, U_9\}$ in the system where each user has some computational tasks which need to be executed on the resources in the system. A total of 25 computational tasks are considered in the system. The distribution of tasks among the users is random. The group preference for each task is uniformly distributed over the interval [0, 4], which corresponds to the five groups in the system. The budget of each task is uniformly distributed over the interval [45 G$, 630 G$]. The lower limit of tasks' budget interval is given by the product of the lowest com-

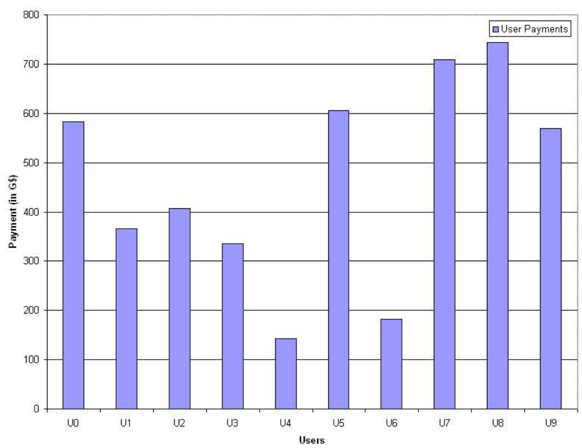

Figure 2. Users payments

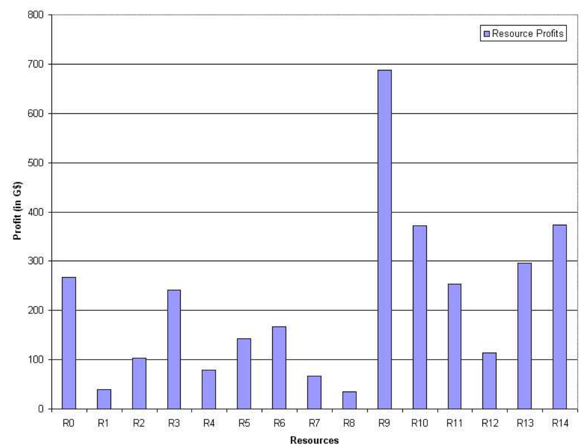

Figure 3. Resource profits

putational time of a task and the lowest reservation price of a resource while the upper limit is given by the product of the highest computational time of a task and the highest reservation price of a resource. The execution time of tasks follows an exponential distribution. The approximation error ϵ in the approximation algorithm for solving the combinatorial auction problem (CAP) is chosen to be 0.5. The combinatorial auction based resource allocation protocol is evaluated by simulation in order to determine its economic efficiency and the system performance.

The *payment* given by a winning user is defined as the price value for which the user wins a bundle in the allocation. The motivation for studying users payments for the combinatorial auction model is to analyze two factors. The first factor is to determine how the optimal allocation solution determined by the protocol affects the price values bidded by the loosing users for the next round. The second factor is to observe the effect of the price values bid by the users on the obtained allocation solution. Figure 2 shows the payments handed by the users in the combinatorial auction based resource allocation protocol.

The results in Figure 2 show that the payments of users U_0, U_1, U_2 and U_4 are half of their total budget as shown in Table 2. This is because they bid half the value of their total budget in the round in which they win. For the other users U_3, U_5, U_6, U_7, U_8 and U_9 the payments are close to the total budget because they bid close to their total budget in the winning round. The bidding strategy of each user is random. The protocol finds an approximate allocation such that more users won in each round ensuring that the same resource is not shared among the winning users in the same round and that the revenue is maximized.

4.2 Resource Profits

The *profit* for a resource is defined as the difference between the payment received from the winning users (whose tasks are allocated to the resource) in each round of the combinatorial auction and the cost incurred in processing the jobs allocated to it. The resource profits are studied to analyze how the protocol favors the resources in terms of profit. Figure 3 shows the profits for the resources in the combinatorial auction based resource allocation protocol.

In the group G_0, resource R_{10} has the highest profit. This is due to the fact that the user U_5 that won the bundle $\{R_{10}, R_{11}\}$ has a high budget as shown in Table 2 and bids a price value close to the total budget due to which the payment handed to R_{10} is the highest among group G_0. Resource R_5 has the lowest profit in the group G_0 because of the allocation of a high computational cost task on R_5. In the group G_1, resource R_1 has the lowest profit because user U_4 that won the bundle $\{R_7, R_1\}$ bids a price value much less than the total budget due to which the payment to R_1 is less. Resource R_{11} appears twice in the allocation due to which the payment of R_{11} is high and and thus the profit is high too. Resource R_{12} has the highest profit in the group G_2 because it appears three times in the allocation. It has a low value because of the allocation of a high computational cost task. Resource R_8 has the lowest profit in the group G_3 and among all groups as well. This is due to the fact that the users U_0 and U_1 that won the bundles containing R_8 bid a price value much less than their total budget due to which payment to R_8 is low and thus the profit is low. Resource R_9 has the highest profit among all the resources. This is because R_9 appears three times in the allocation and each time the winning user bids a price value close to its total budget due to which the payment to R_9 is high and the profit is also high.

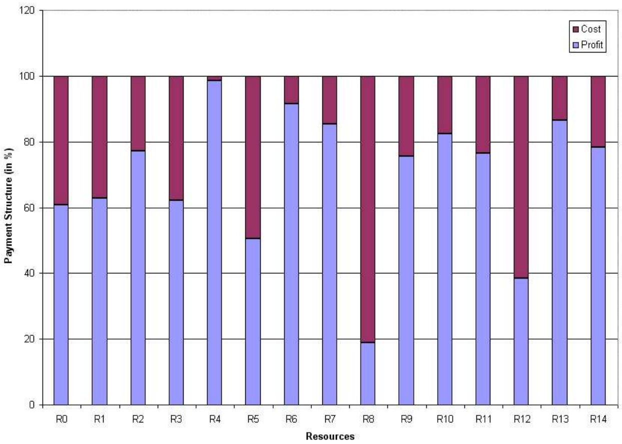

Figure 4. Payment structure for each resource

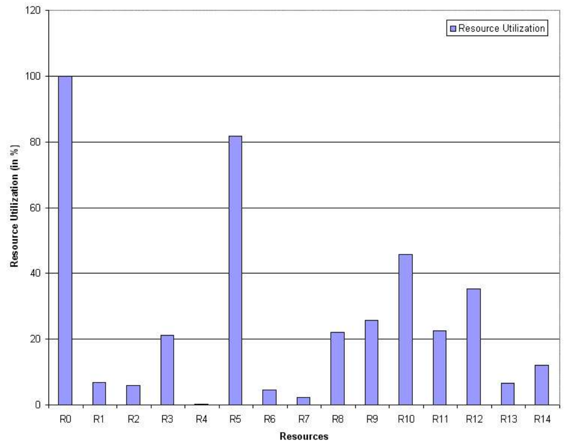

Figure 5. Resource utilization

It can be concluded that the protocol is beneficial for each group of resources and the profit of all the resources as a whole is maximized.

4.3 Payment Structure

The *payment structure* for each resource is defined as the representation of total cost and profit of the resource as fractions of the total payment received by the resource. Figure 4 shows the payment structure of all the resources in the combinatorial auction based resource allocation protocol.

Resource R_8 has the lowest profit of 20% and the highest cost of 80% among all the resources. This is because it appears in the winning bundles of users U_8 and U_1 who bid price value much less than their total budget which makes the payment and the profit low. Further a high computational cost task is allocated to R_8 due to which the cost of R_8 is high. Similar reasons hold for resources R_5 and R_{12}. Resources R_0, R_1 and R_3 have profit close to 60%. This is due to the fact that the winning users for these resources bid price values close to their total budget due to which the payment is high. The cost of these resources is around 40% due to the allocation of average computational cost tasks. The other resources have profit in the range of 75% and 95%. The profit is high because the winning users for these resources bid prices very close to their total budget, thus making the payment high. Resource R_4 has the lowest cost 2% because of the allocation of a very low computational cost task.

It can be observed that the average profit for almost all resources is above 50%. The protocol benefits all the resources in the system.

4.4 Resource Utilization

Resource utilization (ρ_i) for a resource R_i is defined as the ratio of the total execution time ($T_{exec,i}$) at the resource to the total simulation time (T_{sim}): $\rho_i = \frac{T_{exec,i}}{T_{sim}}$. Figure 5 shows the resource utilization for all resources in the combinatorial auction based allocation protocol.

Resources R_0, R_5 have high resource utilization in the range of 80% to 95% because high computational cost tasks are allocated on these resources. Resources R_8, R_9, R_{10}, R_{11} and R_{12} have utilization in the range of 20% to 40%. This is because of allocation of a mix of average and low computational cost tasks. The other resources have utilizations in the range of 5% to 20%. This is because of allocation of low computational cost tasks.

In each round, the protocol allocates tasks uniformly among resources within a group, thus utilizing fairly these resources.

5. Conclusion

In this paper we presented the combinatorial auction allocation model for grids, and proposed a combinatorial auction-based resource allocation protocol. We simulated this protocol using the SimGrid simulation framework and evaluated it in terms of economic efficiency and system performance. In future work we will compare the combinatorial auction based resource allocation protocol with other auction based resource allocation protocols in terms of economic efficiency and system performance. We also plan to develop a prototype allocation system on a real grid environment which will allow real life testing of the proposed resource allocation protocol.

References

[1] D. Abramson, R. Buyya, and J. Giddy. A computational economy for grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computing Systems*, 18(8):1061–1074, October 2002.

[2] A. Andersson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner determination. In *Proc. of the Fourth International Conference on Multi-Agent Systems*, pages 39–46, July 2000.

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

[4] R. Buyya and M. Murshed. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1175–1220, November 2002.

[5] H. Casanova. Simgrid: A toolkit for the simulation of application scheduling. In *Proc. of the IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 430–437, May 2002.

[6] K. M. Chao, R. Anane, J. H. Chen, and R. Gatward. Negotiating agents in a market-oriented grid. In *Proc. of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 406–407, May 2002.

[7] C. Chen, M. Maheswaran, and M. Toulouse. Supporting co-allocation in an auctioning-based resource allocator for grid systems. In *Proc. of the 11th IEEE Heterogeneous Computing Workshop*, pages 89–96, April 2002.

[8] D. Ferguson, Y. Yemini, and C. Nikolaou. Microeconomic algorithms for load balancing in distributed systems. In *Proc. of the 8-th IEEE Intl. Conf. on Distr. Comp. Systems*, pages 491–499, 1988.

[9] I. Foster and C. Kesselman. *The Grid: a Blueprint for a New Computing Infrastructure*. 2nd edition, Morgan Kaufmann, San Francisco, CA, 2003.

[10] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pages 548–553, July 1999.

[11] R. Gagliano, M. Fraser, and M. Schaefer. Auction allocation of computing resources. *Communications of the ACM*, 38(6):88–100, June 1995.

[12] J. Gomoluch and M. Schroeder. Market-based resource allocation for grid computing: A model and simulation. In *Proc. of the 1st International Workshop on Middleware for Grid Computing*, pages 211–218, June 2003.

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

[14] J. Nabrzyski, J. M. Schopf, and J. Weglarz. *Grid Resource Management, State of the Art and Future Trends*. Kluwer Academic Publishers, Boston, MA, 2003.

[15] M. H. Rothkhof, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, August 1998.

[16] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pages 542–547, 1999.

[17] S. Vazhkudai and G. von Laszewski. A greedy grid - the grid economic engine directive. In *Proc. of the 15th IEEE International Parallel and Distributed Processing Symposium*, pages 1806–1815, April 2001.

[18] R. Vohra and S. de Vries. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, Summer 2003.

[19] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. Analyzing market-based resource allocation strategies for the computational grid. *The International Journal of High Performance Computing Applications*, 15(3):258–281, Fall 2001.

[20] R. Wolski, J. S. Plank, T. Bryan, and J. Brevik. G-commerce: market formulations controlling resource allocation on the computational grid. In *Proc. of the 15th IEEE International Parallel and Distributed Processing Symposium*, April 2001.

[21] E. Zurel and N. Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proc. of the 3rd ACM Conference on Electronic Commerce*, pages 125–136, October 2001.