Task-Oriented Computational Economic-Based Distributed Resource Allocation Mechanisms for Computational Grids

Linli He and Thomas R. Ioerger
Department of Computer Science
Texas A&M University, College Station, TX77843-3112
{linli, ioerger}@cs.tamu.edu

Abstract

In computational grids, heterogeneous resources with different ownerships are dynamically available and distributed geographically. It is not realistic to build the resource allocation mechanisms for such computational platform without considering economic issues. Developing computational economic-based approaches is a promising avenue for building efficient, scalable and stable resource allocation mechanisms without a centralized controller for computational grids. The key difficulty in building a computational economic-based resource allocation mechanism is measuring the economic value of resource usage. In this paper, we propose a task-oriented mechanism for measuring the economic value of using heterogeneous resources in computational grids. This mechanism provides feasibility for resource users to evaluate their outsourcing decisions. It also gives resource suppliers incentive to provide their resources to computational grids. Based on this mechanism, auction-based, commodities market-based and game theory-based distributed resource allocation mechanisms are established for computational grids.

1. Introduction

In computational grids, heterogeneous resources with different ownerships are dynamically available and distributed geographically. The users' resource requirements in the grids vary depending on their goals, time constraints, priorities and budgets. Allocating their tasks to the appropriate resources in the grids so that performance requirements are satisfied and costs are subject to their budgets is an extraordinarily complicated problem. Conversely, resource suppliers vary in the resource capability, availability, cost and security policies. Allocating their resources to the proper users so that utilization of resources and the profits generated are maximized is also an extremely complex problem.

From a computational perspective, it is impractical to build a centralized resource allocation mechanism in such a large-scale distributed environment [2]. Providing incentives for both users and resource suppliers to participate in a computational grid is also a big challenge, which must be addressed in the resource and task allocation mechanisms, because it is the key to maintaining the stability of a computational grid. Furthermore, the effective agents (both users and resource suppliers) in computational grids are inherently self-interested because of their different ownerships. Self-interested agents make their own decisions according to their budgets, capabilities, goals and local knowledge, without considering the global good of the entire grid. It is not realistic to build the resource allocation mechanisms for such a grid without considering economic issues. Hence, developing computational economic-based strategies [14] is a promising direction for research into building efficient, scalable and stable resource allocation mechanisms without centralized controllers for computational grids.

Researchers in both high performance computing and multi-agent systems have applied computational economic-based mechanisms for distributed resource allocation problems. Wolski et al [1, 2] and Buyya [3] pioneered investigation of the commodity market-based resource allocation mechanisms in computational grids. Sandholm [4] addressed the fact that the computing, communication, and privacy issues are deeply intertwined with economic incentive issues.

The key difficulty in build a computational economic-based resource allocation mechanism is measuring the economic value of resource usage by a common currency. Previously, people assumed there existed a general currency (denoted by grid dollars [1,2,3]) to measure the cost of using a certain resource. But nobody addresses how to translate the value of using different type of resources into such grid dollars.

This gap keeps computational grids from being realistic because it is difficult to convince users that participating in computational grids is less expensive than purchasing more computational resources while obtaining the same amount of computational power for their computational tasks. Similarly, for resource suppliers, it is hard to evaluate the profit of putting resource into a grid without such a measurement. For both users and suppliers, joining a grid will incur more security and maintenance cost than having only their own computational resources to execute their own tasks. They have to be convinced that the extra cost is worthwhile.

This paper begins with proposing a task-oriented mechanism of measuring the value of resource usage in a computational grid. Based on this mechanism, the values of using heterogeneous resources can be translated into a common currency. Based on the value of resource usage, both computational economic-based and game theory-based distributed resource allocation mechanisms are established for computational grids.

The rest of this paper is organized as follows: Section 2 presents the task-oriented mechanism for measuring the economic value of using resources in computational grids. Based on this task-oriented mechanism, Section 3 gives both auction-based and market-based resource allocation mechanisms for computational grids. Section 4 describes the game theory-based resource allocation mechanism for computational grids. Section 5 gives summary and our future work.

2. Value of Resource Usage

Traditionally, people measure resource usage by the amount of computational units of the corresponding type of resources used. For example, the usage of a CPU is measured by the number of time slots used [1,2]. Currently, people inherited this measuring tradition in computational grids. The price settings are based on computational units of resource usage in the economic-based resource allocation strategies for a computational grid. Based on this price setting mechanism, in order to allocate their tasks to appropriate resource suppliers, users must be aware of the amounts and kinds of resource units they need for a certain task to calculate the cost and predicate the performance.

However, unlike most traditional computational platforms, even the same types of resources in computational grids vary over a wide range of capabilities. The tasks in computational grids usually need intensive resources that have different capabilities. A typical example is a task that involves a large-scale difference in clock rates of CPUs. It is not appropriate to measure the CPU usage only according to the number of time slots required, because the computational capabilities of the time slots of different CPUs used could be different. The prices of CPU units should not only reflect the time slots but also the differentiation of physical capabilities of the CPUs.

Unfortunately, it becomes extremely complex for users to evaluate whether they should outsource a resource intensive task if different CPU units have different prices. Meanwhile, the available resources in a grid could differ at different times. Hence, even identical tasks must be evaluated at each time when they will be executed. Also, different users have different performance criteria and requirements. They would like to find the cheapest resources that can satisfy their performance requirements. It is too complicated to use the value of computational resource units to represent users' different preferences.

Thus, a new mechanism for measuring the economic value of resource usages should be developed to hide the differentiation of physical capabilities of the same type of resources. The value of a resource usage also should reflect the different user preferences in a grid.

2.1. An Observation

For the sake of simplicity, we use CPUs (processors) as our example resource to describe how to model the capabilities of a resource by considering the certain performance it can achieve based on certain tasks instead of using the resource usage units. To begin, we give a simple observation to show the basic idea behind the task-oriented mechanism of measuring the value of a resource usage in this paper.

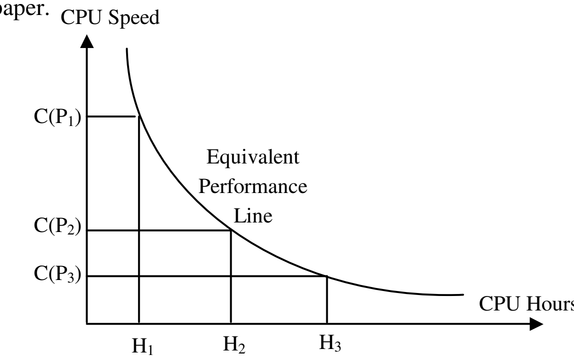

Figure 1: Equivalent Performance by Different Processors

Suppose there are three processors P_1, P_2, and P_3, which have different speeds (Here, we do not specify what exact meaning of the speed of a CPU. It could be measured by MIPs, clock rates, or any other kind of standard units) from the highest to the lowest respectively. Given an identical job, these three processors would finish it in different amounts of time (We assume all other conditions are same, e.g. same amount of RAM associating with each processor). Figure 1 shows the performance of each processor. The equivalent performance line depicts the fact that these three processors P_1, P_2, and P_3 finish an identical job within H_1, H_2, and H_3 CPU hours respectively. The amount of work they did is same, but they used different amounts of time.

If a resource user gives the same job to P_1, P_2, and P_3, how should the processor owner charge the user for using different processors? If all processors can satisfy the deadline of a job, the resource user would prefer not to pay extra for using P_1. But if the deadline of the job is tight, he may be willing to pay more for using P_1. Therefore, in order to set proper prices for using P_1, P_2, and P_3 to execute an identical job, the resource suppliers need to consider both the different capabilities of the processors and the users' performance preferences. Hence, modeling resource capabilities by considering the tasks a resource can accomplish while satisfying the corresponding performance requirements is critical for measuring the economic value of resource usage in computational grids.

2.2. Modeling Resource Capabilities

The main performance criterion of CPUs is how long it takes to finish a job. Therefore, we can define the capability of an individual processor as follows:

Definition 1: Given a task k with duration D and a processor P, the capability of the processor P for executing task k is denoted by CanSingle(P, k, D). CanSingle(P, k, D) is true if and only if P can finish k within D.

If time constraints are the only performance criteria, it is easily to extend above definitions to other types of computational resources such as disks and network bandwidth. In this paper, we only consider the time constraints as the performance requirements for modeling the resource capabilities in grids. The general definition of a group of resource in a computational grid is as follows:

Definition 2: Given a set of task $K = \{k_1, \ldots, k_n\}$ with a duration D and a group of resources $G = \{R_1, \ldots, R_m\}$. R_is in G could be heterogeneous. The capabilities of a group of resources in G for executing the tasks in K is denoted by CanGroup(G, K, D). CanGroup(G, K, D) is true if and only if the resources in G can finish all k_j within D.

These definitions encapsulate the physical differentiation of resources from users and allow users to ignore the physical capabilities of resources. Definition 2 implicitly indicates that there should be some scheduling algorithms for heterogeneous resources, which can schedule tasks in K properly so that the resources in G can achieve tasks in K within D. The quality of a scheduling algorithm has a strong impact on the capabilities of a group of resource, but all these scheduling algorithms are also hidden from users who have tasks in K. The owner of the resources in G will take care of these scheduling algorithms.

2.3. Economic Value of Resource Usages

In this paper, the economic value refers to the true value of a commodity that can generally be accepted by all agents in an economic model. The economic value of a resource usage is not the value of a resource itself but the value of using the resource.

From a user's perspective, no matter what kind of processors are provided to execute an identical task, the economic value of using these processors are equivalent to the user if they can finish the task with satisfying the time constraints of the task. The economic value of using a processor is decided by the task it executes. For individual processors, we have the following claim for the equivalent values of using two different processors:

Claim 1: Given a task k with duration D, and two processors P_i and P_j. P_i and P_j may have different clock rates. The economic value of using P_i to execute k is denoted by V(k, D, P_i). We say V(k, D, P_i) = V(k, D, P_j) if and only if both CanSingle(P_i, k, D) and CanSingle(P_j, k, D) are true.

Similarly, from a user's perspective, the economic value of using a group of resources is decided by the tasks they execute. Hence, we have the following claim (We will not prove these claims as they are common senses.) to address the equivalent value of using two groups of resources:

Claim 2: Given a set of task $K = \{k_1, \ldots, k_n\}$ with a duration D and two group of resources $G_1 = \{R_{11}, \ldots, R_{m1}\}$ and $G_2 = \{R_{11}, \ldots, R_{r1}\}$. The resources in G_1 and G_2 could be heterogeneous. The economic value of using G_1 to execute the tasks in K is denoted by V(K, D, G_1). We say V(K, D, G_1) = V(K, D, G_2) if and only if both CanGroup(G_1, K, D) and CanGroup(G_2, K, D) are true.

An important fact is that any agent in a computational grid does not have the super power to set the true economic value of using a resource. The value should be decided by the interaction among all agents (both the resource users and suppliers) in the grids. Whether an established value of a resource usage is stable depends on the relationship between the amount of resources demanded and supplied. Indeed, the real value of resource usage should be equal to the amount of money by which users are willing to pay for as well as the suppliers are willing to sell. The value of using a resource is not the price of using the resource, but a fair price should reflect the real value of using a resource [14].

Different users have different performance preferences. They search for the cheapest resources that can satisfy their performance requirements based on their local knowledge, computational capabilities and budget limitations. Users do not need to know how many resource units their tasks need or the exact time slots their tasks are executed on each resource. They do not need to be aware the differentiation of resource capabilities, either. They only want to know the resource consumption as a whole for a certain task in order to evaluate the outsourcing decision.

Suppliers try to allocate their resources to the most profitable tasks and prevent their resource from being idle as much as possible. Suppliers plan and schedule their resource based on user requirements, the costs and their local knowledge. Allocating resources to a certain task is a service provided to the corresponding user by a resource supplier. The cost of a service is affected by how much resource it needs, how long it takes, and which levels of customer service are required. Consequently, the resource demand for users turns out to be a service demand to suppliers. Different services result in different costs.

Based upon the above analyses, any mechanism for measuring the economic value of resource usage in computational grids should obey the following principles:

- The economic value of using a resource is evaluated by the task executed by the resource. The value of using different resources with different physical capabilities to execute same tasks can be the same.
- The values of using two groups of resources are equal if they can achieve identical tasks while satisfying the same performance requirements.
- The real economic value of using a resource is established through the interaction among all agents in a computational grid.
- The established economic value of using a resource is stable if the relationship between the amount of resource demanded and supplied is also stable.
- A fair price of using a resource generated through an economic-based resource allocation mechanism can reflect the real economic value of using the resource.

2. 4. Task-Oriented Mechanism for Measuring the Economic Value of Resource Usage

Again, we use CPU as an example to illustrate our mechanism. Referring to Figure 1, three processors with different speeds deal with an identical task. We say the usage of each processor for the task is the same. Mathematically, "the usage of each processor" here refers to the area of each rectangle in Figure 1. The formal definition of the usage of a processor to execute a computational task is given as follows:

Definition 3: Given a task k and a processor P, the speed of P is C(P). P needs H hours to finish k. The usage S(k, P) of P for executing k is the following:

$$S(k, D, P) = C(P) \times H \qquad (2\text{-}1)$$

This definition reflects the amount of processor usage to execute a task no matter what kind of processors are used. The following claim is obviously true:

Claim 3: $S(k, D, P_x) = S(k, D, P_y)$ for given P_x and P_y which are two processors with different speeds.

Claim 3 means the total amount of resource usage for executing the same task is not changed by using processors with different speeds. Based on this claim, we can establish a mechanism to translate the usage of CPUs with different speeds to a common measurement. The idea is to establish a standard speed and convert real CPU usage to the usage of a virtual CPU with the standard speed. For example, assuming other conditions are same, if a 3MHz CPU can finish a task in 3 minutes, then a 1MHz CPU can finish it in 9 minutes (We assume that the task can be divided into 3 subtasks evenly and all other conditions are same). Then 3 1MHz CPUs can finish the task in 3 minutes. Hence there are two directions to convert the CPU usage of a task, one is changing the duration and the other is changing number of CPUs with the standard speed.

Users in computational grids generally expect to finish their tasks as soon as possible. In practice, users would expect their tasks finished in certain duration (e.g., the duration between job submitting time an expected completion time). Given a task with certain duration, we can measure the usage of CPU for executing the task by calculating how many standard CPUs should be used to execute the task while satisfying the time constraints given by the user. We also define a standard time unit to measure the expected duration of a task. Thus, the definition of the usage of processors to execute a computational task is modified as follows:

Definition 4: Given a task k with a certain expected duration D, the standard CPU speed is $C(P_s)$ and the standard time unit is D_s. $D = m \times D_s$. In order to finish k within D, there should be n processors with speed $C(P_s)$ working simultaneously (assume k can be divided into n subtasks evenly) or a processor with speed $n \times C(P_s)$. The usage S(k, n, P) of CPUs for executing k is the following:

$$S(k, D) = n \times C(P_s) \times m \times D_s \quad (2\text{-}2)$$

This definition implies that the CPU usage of any task can be measured through a standard speed and a standard time unit. The standard unit of CPU usage is given by equation 3-3:

$$S_s = C(P_s) \times D_s \quad (2\text{-}3)$$

Based on equation 2-3, the CPU usage for executing a task can be measured through S_s by changing the number of CPUs with the standard speed or the number of the standard time units to finish a task. Thus equation 2-2 becomes:

$$S(k, D) = n \times m \times S_s \quad (2\text{-}4)$$

If the economic value of S_s is V_s, it is easy to calculate the corresponding economic value V(k, D) of CPU usage for executing the task.

$$V(k, D) = n \times m \times V_s \quad (2\text{-}5)$$

However, equation 2-5 does not reflect the common sense that the CPU usage for executing a task in a shorter duration might have higher value that the one for executing the same task with a longer duration. In order to catch this fact, we modify equation 2-5 as follows:

$$V(k, D) = (n+\lambda_1 n_0)\times (m+\lambda_2 m_0) \times V_s \quad (2\text{-}6)$$

Where, n_0 and m_0 refer to the increasing number of CPUs with the standard speed and the number of the standard time units respectively. n_0 and m_0 can be negative. The coefficients λ_1 and λ_2 in equation 2-6 imply that changing the number of CPUs with standard speed and the number of the standard time units to finish the same task results in different economic values of CPU usage. Mathematically, equation 2-6 equivalent to

$$V(k, D) = nmV_s+\lambda_1 m\, n_0 V_s + \lambda_2 n\, m_0 V_s + \lambda_1\lambda_2\, m_0\, n_0 V_s$$

If only either m_0 or n_0 is equal to 0, then we can have:

$$V(k, D) = nmV_s+\lambda_1 m\, n_0 V_s \quad (2\text{-}7)$$

or

$$V(k, D) = nmV_s+\lambda_2 n\, m_0 V_s \quad (2\text{-}8)$$

If $\lambda_1 \neq \lambda_2$, then changing the number of CPUs with standard speed and the number of the standard time units to finish a same task causes different economic values of CPU usage.

We have now established a task-oriented mechanism for measuring the economic value of CPU usage for executing a task in a computational grid. It can be extend to a group of heterogeneous resources through defining a standard computational unit based on the definition of the capability of a group of heterogeneous resources. We leave this for the future work.

3. Economic-Based Resource Allocation

Unlike traditional computational platforms, to build a stable computational grid, the incentives of both users and suppliers to participate the grid must be addressed in the resource allocation mechanisms. Using computational economic-based strategies is a promising direction for building such resource allocation mechanisms in grids [1,2,3].

Economic-based mechanisms have been extensively studied as resource allocation mechanisms for distributed computing systems [13]. However, under the computational grid settings, no such mechanism has been established in real applications. Researchers have done experiments based on some popular computational economic-based mechanisms such as auction and market mechanisms [1,2,3]. Based on the traditional resource usage value measurements, it is hard to convince people that it is worthwhile to join a computational grid instead of purchasing their own new computational resources. We establish economic-based resource allocation mechanisms for computational grids based on the task-oriented mechanism of measuring the value of resource usage.

A question raised here is how to define a user task. Which one is more appropriate, a simple task that can be executed by an individual resource or a complex task that requires the cooperation among heterogeneous resources? In our mechanisms, users decide which kind of task they want to outsource. If a user decomposes a complex task into a group of simple tasks and outsources them, he has to take care of scheduling tasks to satisfy the time constraints of the complex task. If a user wants to outsource a complex task entirely, resource suppliers do the scheduling and bid for the task if they can satisfy the time constraints.

3.1. Auction-Based Mechanism

Recently, combinatorial auction-based resource allocation mechanisms have extensively explored [1,2,3,4,5,9,13,14]. It is the simplest economic-based resource allocation mechanism in the sense that the implementation of an auction-based mechanism is simpler than that of other types of economic-based mechanisms.

Wolski et al. [1, 2] have showed that the performance of auction-based resource allocation mechanisms is beaten by market-based resource allocation mechanisms. However, their experiments base the resource usage measurement on setting the bids for the computational units of resources. This is the most important reason that the auction-based resource allocation mechanisms have lower performance than the market-based mechanisms. The tasks in a computational grid generally require intensive resources. If all resource units needed are obtained by auctions, the scheduling process must start after the user ends his bid on resources. Also, finding the winner in a combinatorial auction has been approved a NP-hard problem [5]. As a computational platform, it is hard to improve the efficiency of resource allocation through auction.

However, based on our task-oriented mechanism for measuring resource usage value, the commodities are not the usage units of the computational resources but the tasks from users. It is valuable to re-examine the performance of the auction-based resource allocation mechanism under this new setting. The auctioneers are users who have computational tasks. When a user initializes an auction, he has already set the performance requirement. The resource suppliers bid for the opportunity to provide resources that are capable of executing users' tasks by giving a certain cost. A resource supplier has already warranted the performance when he commits a bid. The user will select the auction winner to execute the task.

3.2. Market-Based Mechanism

Market-based strategies seem naturally appropriate for building economic-based resource allocation mechanisms in computational grids, because the users and the resource suppliers in computational grids are easily translated into buyers and sellers in commodity markets. Intuitively, the commodities are usages of resources such as CPU time slots, number of files stored etc. This is probably the reason that the original computational-based resource allocation mechanisms were built on setting prices for the units of resource usage. However, in real applications, it is impractical for users to know the physical capabilities of CPUs that are available in a computational grid at certain time in advance. It is infeasible for users to predicate how many computational units of a certain type of resources they need to execute their tasks.

We propose a market-based resource allocation mechanism for computational grids, also based on our task-oriented mechanism for measuring resource usage value. Again, the commodities in markets are not the usage of the computational resources. Resource suppliers provide their resource to execute users' tasks as services to the users. That is, resource suppliers sell services to users instead of selling the amount of resource usages. Users do not need to know the amount of resources required to execute their tasks exactly. They only need to know the cost of the services provided by the resource suppliers. A stable price for a service should be established in a long run through the interactions among all agents in grids. This stable price should be fair in the sense that it reflects the real economic value of resource usages required for the corresponding service. An important question we need answer here is how to set the original prices for services provided by resource suppliers. There are two possible approaches.

In the first approach, resource suppliers set the original prices for the services they provide based on the total amount of resources required, the qualities of services and the expected profit they want to make. Note that resources used to provide the same service could be different at different times. Consequently, the amount of resource required does not reflect the real value of the resource usage of the service. In fact, a supplier should set the original price for a service based on the task that the service can finish while satisfying the performance requirement.

Setting the appropriate original prices is very hard for resource suppliers, because nobody knows the real economic value of a resource usage at the beginning. The second approach combines auction-based and market-based mechanism.

3.3. Combined Auction-Based and Market-Based Mechanism

As Wolski et al [2] pointed out, commodity markets and auctions represent two ends of a spectrum of market formulations, from satisfying all bidders and sellers at a given price to satisfying one bidder and one seller at a given price. It is obviously possible to consider market organizations that are between the extremes. We construct such a combinational resource allocation mechanism by using auctions for a certain task at first. The original price of a service to achieve the task can be set. Then, put the prices in perfect competitive markets [2], and let the relationship between service demand and supply and the interaction among agents adjust the prices to be fair.

3.4. Market-Based Mechanism Evaluation

Wolski et al [1, 2] have proposed four criteria for evaluating economic-based resource allocation mechanisms in computational grids: the grid-wide price stability, market equilibrium, application efficiency and resource efficiency. Price stability ensures scheduling stability. Equilibrium measures the fairness of prices. Application and resource efficiency measures how well the grid functions as a computational platform. These four criteria should also be used to evaluate our auction and market-based resource allocation mechanisms for grids. We have not done this evaluation yet.

Furthermore, we should consider the individual rationality [12] of each individual agent in a grid. If a user would spend more money by outsourcing tasks than purchasing more resources, that user will not outsource the tasks. Similarly, if a supplier cannot make profit from providing resources to a grid, the supplier will not join the grid. Although this is the most important criterion for giving incentives to agents participating in a grid, it is missed for evaluating an economic-based resource allocation mechanism in previous work [1,2,3].

We have not finished the evaluation results, so we cannot include them in this paper.

4. Game Theory-Based Mechanism

The most obvious weakness of the commodity market-based mechanism for resource allocation mechanism in computational grids is that is no such real market exists currently. It is hard to verify that the empirical results in experimental settings can be duplicated in a real market for computational grids.

Another kind of economic-based resource allocation mechanisms is based on Game theory. They do not need price setting mechanisms for support. The typical example is coalition formation. Self-interested agents form a coalition to pool their capabilities and resources to solve their own problems more efficiently and less expensively. Researchers in multi-agent systems society have developed quite a few coalition formation-based resource allocation mechanisms for cooperation among self-interested agents [11]. According to our knowledge, this type of economic-based resource allocation mechanisms has not been investigated for computational grids yet.

4.1. Coalition Formation-Based Mechanism

Indeed, coalition formation is a more practical mechanism for resource allocation in computational grids compared to other economic mechanisms. The reason is the following: The tasks in a computational grid need intensive computational resources in real applications generally. The potential users who have such tasks are not individuals but organizations that already have owned a large amount of computational resources. Even so, these users need more computational power to execute their tasks at some peak times. Meanwhile most of their resources are idle at other times. In other words, agents in computational grids could be both resource users and suppliers. Coalition formation provides a cooperation mechanism for multiple organizations that have this problem to put their resources into a grid and satisfy their resource requirements at peak time without purchasing more resources as well as improve their resource utilization.

Indeed, real organizations have already started to negotiate with each other to share their computational resources to execute large computational tasks. For example, some research labs negotiate with commercial companies to use the computers of those companies at night from 6pm to 7am and the commercial companies can use the machines of the research labs whenever they are idle. Human subjects execute the negotiation processes to establish certain resource allocation mechanisms for such a computational grid currently.

However, these negotiation processes only can be run to initialize a computational grid. Afterwards, if there are internal or external changes (e.g. software upgrades) for the grid, the resource allocation mechanisms should be changed by resetting some other negotiation processes. Furthermore, negotiations among human subjects cannot be in task level because it is impossible for human subjects to negotiate for every computational task. Hence, it is impractical to build efficient and adaptive resource allocation mechanisms for computational grids through negotiations among human subjects. Automatic distributed coalition formation mechanism through automate negotiation should be established for resource allocation.

4.2. Distributed Coalition Formation

Game theorists did not provide algorithms for forming coalitions. Researchers in multi-agent systems society have been developing algorithms for forming buyer coalitions in electronic markets so that buyers can obtain greater discount from sellers without purchasing more than they really want to buy [8,9,10]. In the computational grids, agents can achieve desired performance by forming coalitions to share computational resources without purchasing more resources individually.

Buyer coalition formation generally involves a group leader who is responsible for evaluating the value of each possible coalition formed by a group of agents [8,9]. However, in computational grids, a centralized coalition formation mechanism is impractical for allocating resources. Agents are leaving and joining a grid randomly. A centralized resource allocation mechanism is not appropriate for handling the dynamics in such a large-scale environment. A distributed coalition formation mechanism [11] is required, meaning no such group leader exists. Coalitions are formed through negotiation among agents.

Agents evaluate the value of a possible coalition and the benefit they can obtain from joining the coalition based on their own preferences and local knowledge. The value of a coalition is defined as the sum of the values of resource usage required to execute all tasks that the coalition can achieve. An agent will join the best coalition that it finds. The best coalition is the one by which the agent could let all his tasks achieved with minimum cost as well as maximize his own resource utilization.

To develop such a distributed coalition formation mechanism, two important issues should be considered. One is the automatic negotiation mechanism. The other is evaluating the fairness of the payoff division of a coalition.

4.2. Automate Negotiation Mechanism

There are three main issues in defining a negotiation mechanism: the space of possible deals, the negotiation process, and the negotiation strategy [6, 7]. The negotiation invoked for forming a coalition is generally not just two parties in a grid. The space of possible deals for each agent is the total number of the possible coalitions he could join. Obviously, there could be multiple agents involved in one negotiation process. The negotiation strategies could vary based on the preferences of agents. The resulting deals should achieve the Pareto optimality [6, 7], meaning no agent can improve its own utility without lowering the utilities of others. The utility of joining a coalition for each agent is the difference between the cost of executing its tasks by using its own resources and the cost resulted by joining a coalition to execute his tasks using resources in the coalition.

4.3. Payoff Division Evaluation

In order to form a stable coalition in computational grids, the payoff division within the coalition must be fair. We use the core concept in the theory of coalition formation to evaluate whether a payoff division is stable [12]. Namely, any subset of agents in a coalition can get at least as much by joining the coalition as the value of the coalition formed by the agents in this subset. The fairness of the payoff division is critical for the stability of a coalition in a grid. It provides the incentive for agents to stay in the coalition.

5. Conclusion and Future Work

Traditionally, both centralized or decentralized resource management mechanisms for computational platforms have been constructed from the top down, namely, fixed decision rules are imposed to handle all possible situations in resource management in that platform. This design philosophy does not work well in computational grids, because there does not exist an omniscient designer who can develop such a resource management mechanism that satisfy the preferences of all self-interested users and resource suppliers as well as maximize the global efficiency of a grid as a computational platform. Instead, in computational grids, resource management mechanisms should be established from the bottom up, meaning that every resource user or supplier makes individual decision based on local knowledge and preferences without considering the global good, but the global efficiency is generated from bottom up through the interactions among agents [14]. The computational economic-based approach is a promising avenue for building such a distributed resource management mechanism for computational grids.

The key difficulty in building a computational economic-based resource allocation mechanism is measuring the economic value of resource usage by a common currency. This is one of the most important obstacles to make computational grids realistic because it is difficult to convince agents that participating in computational grids is less expensive than purchasing more computational resources to obtain the same amount of computational power.

In this paper, we proposed a task-oriented mechanism for measuring the economic value of using heterogeneous resources in computational grids. This mechanism provides feasibility for grid users to evaluate their outsourcing decisions. It also gives resource suppliers incentive to provide their resources to computational grids. Based on the economic value of resource usages, both computational economic-based and game theory-based distributed resource allocation mechanisms are established.

Currently, we have just establish these economic-based resource allocation mechanisms. To implement all these mechanisms, the appropriate resource discovery mechanisms, negotiation protocols and resource planning and scheduling algorithms are developed to support the entire allocation mechanisms.

Another important issue to consider in the future is how the computational capabilities of agents may affect their resource allocation decisions in computational grids. The complexity of evaluating a resource allocation decision is generally intractable. The bounded rationalities [4] of agents in computational grids should be investigated.

6. Acknowledgement

This work was supported in part by MURI grant #F49620-00-1-0326 from DoD and AFOSR.

7. References

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

[2] Wolski, R., Brevik, J., Plank, J., and Bryan, T., Grid Resource Allocation and Control Using Computational Economies, In Grid Computing: Making the Global Infrastructure a Reality Berman, F, Fox, G., and Hey, T. editors, Wiley and Sons, pp. 747--772, 2003.

[3] Buyya, R. Economic-Based Distributed Resource Management and Scheduling for Grid Computing. Ph.D. Dissertation, 2002.

[4] Sandholm, T. Making Markets and Democracy Work: A Story of Incentives and Computing, in Proceedings of IJCAI, 2003.

[5] Sandholm, T., Suri, S., Gilpin, A., and Levine, D., Winner Determination in Combinatorial Auction Generalizations. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2002.

[6] Rosenschein, J. S. and Zlotkin, G., Rules of Encounter: Designing Conventions for Automated Negotiation among Computers, The MIT Press, 1994.

[7] Kraus, S., Strategic Negotiation in Multiagent Environments, The MIT Press, 2001.

[8] Yamamoto, J. and Sycara, K., A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces, in proceedings of the Fifth International Conference on Autonomous Agents, 2001.

[9] Li, C. and Sycara, K., Algorithm for combinatorial coalition formation and payoff division in an electronic marketplace, in proceedings of the First International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2002.

[10] Lerman, K. and Shehory, O., Coalition formation for largescale electronic markets, in proceedings of the International Conference on Multi-Aent Systems, 2000.

[11] Shehory, O. and Kraus, S., Feasible formation of coalitions among autonomous agents in non-super-additive environments, Computational Intelligence, vol. 15(3), 218-251, 1999.

[12] Kahan, J.P. and Rapoport, A., Theories of Coalition Formation, Lwarence Erlbaum Associates, Inc., London, 1984.

[13] Clearwater, S.H.(Editor), Market-Based Control: A Paradigm for distributed resource allocation, World Scientific, 1996.

[14] L. Tesfatsion., Agent-based computational economics: Growing economies from the bottom up. Artificial Life, 8:55--82, 2002.