Grid Processor Scheduling based on Game Theoretic Approach

Sara Forghanizadeh

CPSC532A Final Project, Fall 2005

Abstract

Grid computing uses the resources of many separate computers connected by a network to solve large-scale computation problems. In this survey, four Grid Computing systems that have a game theoretic approach in processor scheduling have been described. An analysis and comparison of these systems in terms of several important game theoretic attributes has been presented.

1 Introduction

Grid computing uses the resources of many separate computers connected by a network to solve large-scale computation problems [2]. The SETI@home project [1], launched in 1999, is a widely-known example of a very simple grid computing project. SETI is an acronym for the Search for Extra-Terrestrial Intelligence. It uses the processing power of thousands of Idle CPU's that are connected to the internet. In this system, the CPU power is donated by the users who are considered truthful, and there is no competition between them. But this problem becomes more challenging when resource owners are being paid, or other issues exist such as the social reputation gained by participating in the Grid society. There are several Grid computing systems that have a game theoretical approach, such as Popcorn [7], CompuP2P [5], and Nimrod-G.

To our knowledge, there is no game theoretical comparison of such Grid Computing systems. In [3], a comparison of a few market models for Grid Computing is presented, but with an economic perspective rather than a game theoretic one. In this survey, four Grid Computing systems with game theoretical approach are described. The first system named POPCORN [7] uses single and double auction schema for job scheduling. The second one is CompuP2P [5] which is an architecture for sharing computing resources in peer-to-peer networks. The third one is a job allocation system in mobile Grids [4], and uses a bargaining mechanism; and the forth system [6] offers a hierarchical structure and an optimal solution for Grid computing. Analysis and comparison of these four systems is presented according to several game theoretic attributes that are important in Grid Computing systems.

Section 2 describes the structure, the games and the evaluation in each of the four systems. Section 3 presents the analysis and comparison. Section 4 provides a number of open issues in this area.

2 Grid Processor Scheduling Research

This section provides a description of four Grid Computing systems that are based on a game theoretic approach.

2.1 O. Regev and N. Nisan, The POPCORN Market

2.1.1 Structure

There are three main types of entities in POPCORN: 1. The CPU-time buyer 2. The CPU-time seller 3. The market which matches buyers and sellers. These entities communicate via internet. The basic goods that can be traded are either JOPs (Java Operations), or computelets (sub-computations).

2.1.2 The Game

The authors believe the two important properties of the mechanism that should be used is 1. the allocations should be economically efficient (maximize the global utility) 2. they should motivate the buyers and sellers to reveal their true utility of the CPU time. In order to gain the second property, an electronic currency named *popcoin* is defined. Three auction mechanisms are suggested:

1. A repeated Vickery (Second Price) auction: Truthfulness is the dominant strategy in a single Vickery auction, but for a repeated game this condition does not hold. If the buyers and sellers are truth-tellers and if the sellers are homogenous, then this mechanism maximizes the welfare of the economy, but in reality they are not.

2. The simple double auction mechanism: There is no dominant strategy equilibrium in this mechanism. It is not a direct revelation mechanism and is not necessarily Pareto efficient.

3. The repeated Clearinghouse Double Auction: It is similar to simple double auction, but each time more than one buyer-seller pair is found. This is done by finding intersections on current demand-supply curves on fixed time intervals. The equilibrium price is calculated this way, and that single price will be used for all transactions in that round. There is no dominant strategy in this case.

2.1.3 Evaluation

In the simulations, the environment consists of truth-telling buyers and sellers. The Clearinghouse Double Action mechanism showed the highest social efficiency, and the highest stability in terms of the price.

2.2 R. Gupta and A. K. Somani, CompuP2P

2.2.1 Structure

CompuP2P is an architecture for sharing of computing resources in peer-to-peer networks.The network model uses Chord for addressing and node's connectivity, but it is claimed that the offered solution could be easily generalized to other network structures (figure 1). No centralized authority is needed to govern nodes' behavior. Some nodes in the network take the responsibility of being market owner (MO).The network is dynamic as peers join and leave at unpredictable times. There is some electronic payment mechanism to motivate nodes to contribute. It is assumed that peers are selfish, but not malicious.

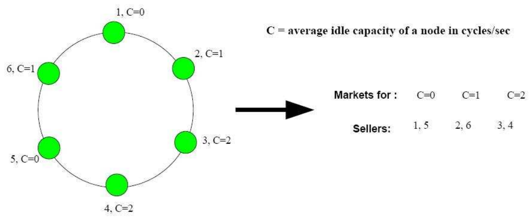

Figure 1: Creating of markets for CPU cycles in CompuP2P

2.2.2 The Game

Different nodes have different amounts of computing power to sell or purchase. Important issues that should be considered in adopting a schema for the creation of these markets in peer-to-peer systems are their simplicity to implement, avoiding the possibility of bottleneck and fast lookup speed.

The search mechanism for the compute power is follows: A client contacts any of the known MO's and forwards the lookup request to it. The selected MO searches for an appropriate market for the desired compute power and returns the IP address of the MO that runs the market of that compute power. That MO is then contacted and provides information on the sellers listed in the market. Nodes have incentive to become MO's since they make profit by charging listing price (LP) from sellers (and/or buyers). There are different pricing schemas that can be used by a MO: (1) Fixed listing price: All sellers are charged the same amount. This is a simple strategy but it seems to be unfair. (2)Variable listing price: a MO charges the buyers on the basis of the market characteristic, e.g. some percentage of the selling price. This schema seems fair to both sellers and buyer, but it is difficult to implement.

There are different pricing strategies which provide incentive to sellers to participate in a trade. The paper defines two different pricing strategies depending on a listing price that is used to compensate a MO. (1) In fixed listing pricing, a MO has no incentive to cheat; so the technique used in Vickery (Second Price) auction can be used. (2) In variable listing pricing the previous schema cannot be employed as it is designed for non-selfish auctioneers (MO's). In order to deal with selfish MO problem, a max-min payoff strategy is proposed. This strategy makes the payoff to a seller and MO complementary to each other. The proposed payoff to the MO and the selected seller are as follows:

$$Payoff_{MO} = (MC'_n + MC'_1)/(MC'_n)^2 Payoff_{seller} = (MC'_1 + 1 \qquad (1)$$

MC'_n and MC'_1 refer to the marginal cost values of the lowest and highest cost supplier as reported by the MO to the buyer. The above payoff values guarantee that the total cost to the buyer is bounded and the best strategy for the MO is to return the lowest cost supplier only. But the MO can still increase his payoff by providing a modified price list.

2.2.3 Evaluation

A Java based prototype of the proposed CompuP2P architecture has been implemented.

2.3 P. Ghosh et al., Job Allocation in Mobile Grids

2.3.1 Structure

This research offers a solution specific to mobile devices such as laptops and PDA's connected to wireless servers. A group of mobile devices are connected to each Wireless Access Point (WAP) Server, and there is a pool of these WAP servers in the Grid Community.

2.3.2 The Game

The problem is modeled as a bargaining game between each mobile device and the WAP server (Figure 2), to determine the pricing strategy which is then used to effectively distribute jobs to mobile devices. Assuming that there are n mobile devices under a single WAP server, this server has to play n such games with the corresponding devices to form the price per unit resource vector, p_i.

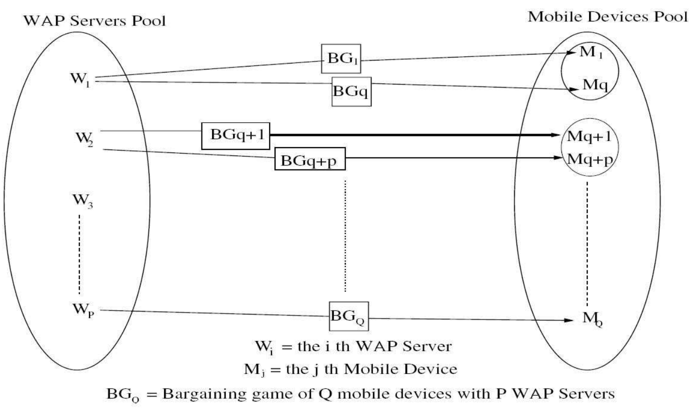

Figure 2: Dynamics of mobile user interactions with different WAP

Both the mobile device (seller) and the WAP server (buyer) have reserved valuations for the CPU cycles, and in order to produce a solution space for bargaining, the individual valuations of these two should overlap. There is a market price M_x for the dealing resource, which depends on the history of interactions, time deadlines, etc. The expected utility functions for the two players are as follows:

$$EU = (reserved\ val\ of\ w - standardprice) \times prob(stabdardprice) \times e^{-z_w t} \tag{2}$$

for the WAP server, and for the mobile device it is as follows:

$$EU = (standardprice - reserved\ val\ of\ w) \times prob(stabdardprice) \times e^{-z_w t} \tag{3}$$

$e^{-z_w t}$ is the discount factor and makes the players come to a conclusion faster, or their expected utility will decrease along the time.

The game starts when one of the players offers a price. This offer is estimated according to the player's computational state, which helps him guess the price the other player might accept. The other player can do three things: Accept the offer and then the job will be done, reject the offer and offer a new price (in the case of the WAP server, a lower price, in the case of the mobile device, a higher price).

Each mobile device is considered as an M/M/1 queue. β_i is the average job arrival rate, μ_i is the average processing rate, k_i is a constant that maps the execution time to the amount of resources and n is the number of mobile devices connected to the WAP server. Φ is the total job arrival rate at the WAP.

Theorem: The overall job execution cost is minimized when

$$\beta_i = \mu_i - \sqrt{k_i p_i \mu_i} \frac{\sum \mu_j - \Phi}{\sum \sqrt{k_i p_i \mu_i}} \tag{4}$$

An optimal algorithm named PRIMAL (Price based optimal) for job allocating according to the above findings is offered. First they sort the mobile devices in decreasing order of their average processing rates ($\mu_i > \mu_{i-1} > ... > \mu_1$). Then they minimize the whole cost by assigning optimal value in equation 4 to β_i to a number of the mobile devices, and assigning $\beta_i = 0$ to the others.

2.3.3 Evaluation

For comparison purpose, two other load-balancing schema besides PRIMAL have been implemented. COOP (cooperative load balancing) scheme minimizes the expected execution time of jobs from mobile device perspective. OPTIM (optimal static load balancing) scheme minimizes the overall expected execution time of jobs from a system, i.e,, the grid user perspective. It is shown that PRIMAL outperforms COOP when the total arrival rate to aggregate processing rate of the system is above 70%.

2.4 Y. Kwok et al., A Hierarchical Structure

2.4.1 Structure

This research suggests a hierarchical game theoretic Grid model, which is claimed to be nearer to reality and makes managing the scalability easier. Figure 3 shows this hierarchy, which consists of three levels: global scheduling level (and there is a centralized global job scheduler), inter-site level, and intra-site level. In other words, the whole Grid system consists of several Grid sites. In each site, there is a local dispatcher who is responsible for declaring the "execution capabilities" of that site to the global job scheduler.

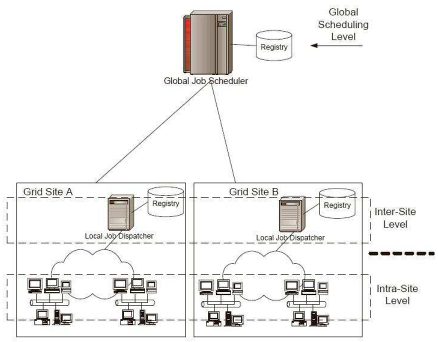

Figure 3: enter your caption here

2.4.2 The Game

Three different games can be defined in this system:

1. Intra-Site Job Execution Strategies: Determines what strategies the selfish computers should take inside a Grid site to maximize their utility. It is assumed that each computer is selfish in the way that it only wants to execute jobs from local users, but does not contribute to the execution of remote jobs.

2. Intra-Site Bidding: Each computer in the site declares its "execution capabilities". The local dispatcher moderates these amounts and sends a single value to the global scheduler.

3. Inter-Site Bidding: The global scheduler should allocate jobs according to the values sent by local dispatchers.

They have focused on the first game so far, and the second and third games are going to be a part of their future work.

As each computer prefers to do local rather than remote jobs, the utility function of each computer is defined as $U_i = \frac{P_i^t}{P_i^r}$. P_i^t is the number of processors in player i, and P_i^r is the number of a sub-group of these processors that are being used for remote jobs. ($P_i^r > 0$ because for each computer there is a remote overhead for being a part of the Grid, such as monitoring the Grid status). The point is, each computer wants to minimize P_i^r, but on the other hand, the Grid site as a whole would like to maximize its Reputation Index(RI) which quantifies the contributions of the site. The problem is how to balance these two.

There is a time deadline for each job. If the job is not done by the deadline, the Grid rejects the job and its reputation decreases. If it is done, the reputation increases. Suppose s_i is the mixed strategy for each computer M_i, which is the probability that the assigned job is taken by him. α is called the *selfishness penalty factor* which quantifies the amount of extra resources incurred should the machine refuse to take up the job earlier. As the time passes, α increases because the deadline becomes closer and more processors will be needed to finish the job. It is proved that for the two-round game and with two players, if

$s_2 < \frac{\alpha}{\alpha+1}$, $\frac{\delta U}{\delta s_1} > 0$, so computer 1's best strategy is always to take the job. If it is less than that, it is best never to take the job. If equal, it would be indifferent. It is shown that the strategy combination (s_1,s_2)= ($\frac{\alpha}{\alpha+1}$,$\frac{\alpha}{\alpha+1}$) is Nash Equilibrium. It is also shown that the optimal strategy is when a symmetric value of s_i is enforced, and a formula for calculating this s_i is provided. It is shown that with the enforced s_i value $\hat{s_i}$, which could be declared to the computers by the local dispatcher, the optimum result will be gained. It is shown that these calculations can be extended to n players and n rounds of the game.

2.4.3 Evaluation

It is shown that the Optimal strategy consistently outperforms the Random and Nash strategies, and the Nash strategy is very poor (close to Random).

3 Analysis and Comparison

This section presents the analysis and comparison of the four systems introduced in previous section. Table 3 shows a comparison of these systems according to the attributes that are explained below.

	POPCORN (1)	CompuP2P (2)	Mobile Grids(3)	Y. Kwok et al.(4)
Decentralized	No	Yes	No	No
Social Efficiency	Yes	No	Unclear	Yes
Maximize Node Utility	No	Yes	Yes	Yes
Handle Selfish Nodes	No	No	Yes	No
Individual Rationality	Yes	Yes	Yes	No
Price Stability	Yes	Yes	Yes	N/A
Communication Efficiency	No	No	Yes	Yes

Decentralized: Whether the system is independent of a central server or coordinator. Though (3) and (4) are network of networks, in the end they depend on a central server. They could be called semi-centralized.

Social Efficiency: (economic efficiency) Maximizing the Grid overall utility, or minimizing the Grid computation cost. All these systems somehow try to maximize the Grid social efficiency. However, in (1), if nodes are not truthful in auctions, the global utility might not really be maximized. (2) is not social efficient because if the fixed-price mechanism is used by the MO, large prices could be paid for small resource needs; and in the second mechanism, MO's can lie in order to increase their own payoff, and consequently increase the global computation cost. In (3) it is not clear how jobs are assigned to different WAP servers; so the author of this report believes that social efficiency cannot be determined in this system. (4) is social efficient, but only in the case that local dispatchers can enforce the optimum mixed strategy.

Maximize Node Utility: All the mechanisms provide ways to maximize individual node utilities. Only in (1) the node utility might not be maximized, and that is because nodes could increase their utility by lying in the auctions, while this system is based on truthfulness.

Handle Selfish Nodes: (1) does not handle the selfish nodes completely, especially it assumes the central server is trusted and not selfish. In (2), though selfishness of MO's is taken care of to some extent, MO's can still decrease social efficiency by declaring wrong price lists to maximize their own payoffs. In (3), nodes cannot harm the system by being selfish, because there is a bargaining mechanism, and if they declare a higher amount of cost, other nodes will be chosen. In (4), there is no good reason for selfish nodes to participate in computing. And besides, the case where local dispatcher and global schedular are selfish has not been taken into account.

Individual Rationality: Whether it is better to participate. In (1),(2) and (3), there is some sort of electronic currency and this is an incentive for the nodes to contribute. In (4), there is no payment for the task (other than increasing the reputation of the Grid site, but not the individual computer); so nodes do not have enough motivation for participating, especially if they are selfish.

Price Stability: Whether the standard deviation of prices remains stable at various supply levels. In (1) the simulation results have shown price stability in repeated double side auction. In (3), as there is a bargaining mechanism, and the offered prices are based on the history of previous games, prices tend to become stable. In (2), as there is competition between MO's after all, the prices will probably tend to be stable in the end.

Communication Efficiency: Whether there is little communication overhead. In (1), it is efficient in repeated vickery auction, but simple and repeated double auctions have communication overhead. (2) is not efficient; the downside of decentralized P2P systems is communication overhead. (3) and (4) use discount factors in order to minimize communication.

These systems can improve the mentioned properties in different ways. For example, if (1) (POPCORN) used a discount factor in double sided auctions, it could decrease the Communication Complexity of these auctions. One of the flaws in (3) (Mobile Grids) is that it is not determined how jobs are distributed between several WAP servers at the first place. Is there a centralized system that performs this task? This could be another challenging game that has not been taken into account.

4 Open Issues

There are several issues in this area that are wide open research problems. For example, Selfishness has not been covered perfectly in current research. Another example is, in the current research there is no difference between the resource scheduling algorithms in the Grids, no matter this resource is a CPU or Storage; this difference is something that can be worked on. As another example, to our knowledge the possibility of malicious nodes in the network has not been covered yet in current systems. As Grid computing has practical applications at the present, it provides an interesting area for doing research.

References

[1] SETI@home: http://setiathome.ssl.berkeley.edu

[2] Wikipedia: http://www.wikipedia.org

[3] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger: Economic Models for Resource Management and Scheduling in Grid Computing, *Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE)*, Wiley Press, USA, May 2002

[4] P. Ghosh, N. Roy, K. Basu and S.K. Das : A Game Theory based Pricing Strategy for Job Allocation in Mobile Grids , *Proc. of the 18th IEEE International Parallel and Distributed Processing Symposium (IPDPS 04)*, Santa Fe, New Mexico, 2004

[5] Rohit Gupta and Arun K. Somani: CompuP2P: An Architecture for Sharing of Computing Resources in Peer-to-Peer Networks with Selfish Nodes, *Proceedings of Second Workshop on the Economics of Peer-to-Peer Systems*, Harvard University, June 2004

[6] Y.-K. Kwok, S. Song, and Kai Hwang: Selfish Grid Computing: Game-Theoretic Modeling and NAS Performance Results, *Proc. IEEE Int'l Symp. on Cluster Computing and the Grid (CCGrid 05)*, Cardiff, UK, May 9-12, 2005

[7] O. Regev and N. Nisan: The POPCORN Market - an Online Market for Computational Resources, *ICE 1998*

[8] M. Senior and R. Deters: Market Structures in Peer Computation Sharing, *Second International Conference on Peer-to-Peer Computing (P2P'02)*, 2002