2008 International Symposium on Nonlinear Theory and its Applications
NOLTA'08, Budapest, Hungary, September 7-10, 2008

A Game Theoretic Approach for User Participation in Grid Projects

Igor Mishkovski†, Dimitar Trajanov†, Sonja Filiposka† and Ljupco Kocarev‡

† Faculty of Electrical Eng. and Information Technology, University Ss. Cyril and Methodious, Skopje, R. Macedonia
‡ Macedonian Academy for Sciences and Arts, Skopje, R. Macedonia
Email: igorm@feit.ukim.edu.mk, mite@feit.ukim.edu.mk, filipos@feit.ukim.edu.mk, lkocarev@feit.ukim.edu.mk

Abstract– In this paper we address the issue of user cooperation in grid projects. We assume that the users are rational, i.e. their actions are strictly determined by self interest and not by the global worthy goal. With this paper we propose a way of attracting and preserving the nodes in the grid environment taking into account the expected benefit of the node and the grid, while incorporating the network influence.

1. Introduction

Potential grid applications allow nodes to accomplish specific job beyond an individual computing capacity. One of the issues in the grid environment is when it is better for the node to cooperate with the environment or to run the jobs by itself.

The grid computing today relies on users' perception of contributing towards a worthy goal by making their idle CPU cycles available for scientific research [1]. A good example is the SETI@Home project [2]. However, in spite of the worthy goal, only a small fraction of users use this type of projects in order to contribute to the planet. This is maybe because the benefit they get is not palpable in the moment. Thus, users often choose not to or simply forget to cooperate.

We introduce a way to change this user attitude by offering him an instantaneous benefit for his participation in the project. We express this benefit as a chance for a user to complete certain jobs for a shorter time. This could be accomplished if the grid environment offers some computation power to the users as the users offer a fraction of their computation power to the grid.

As a tool for modeling the interaction of the node and the grid we propose *game theory* [3], where the node and the grid are two strategic, rational players who want to capitalize on their profit (i.e. minimize their execution time) by participating in the game.

In this paper we spotlight when it is better for a rational node to collaborate in the grid and perform certain computational jobs in order to complete its jobs for a shorter time and by that to help the grid to attain better performances. Knowing that the grid environment wants to attract more nodes, it must offer a certain fraction of the resources it possesses. After attracting a node the grid must try to preserve the node in the grid. Given these assumptions we are able to determine how much the grid must offer from its resources in order to attract some portion of the computing capabilities of the node. Then we calculate under which circumstances the node will continuously offer a portion of its computing power.

The rest of the paper is as follows. In Section 2 we present the related work in using a game theory as a mathematical model for nodes involvement in grid computing. Section 3 depicts the game for node participation in grid environment, while Section 4 identifies the players and their expectation of the game. In Section 5 the extensive game for attracting a node is presented with the strategies of the players and the Nash Equilibrium solution is offered and Section 6 elaborates the game in which the grid wants to preserve the node in the grid. Section 7 concludes and presents future work in this field.

2. Related Work

There are very few papers that observe non-cooperativeness of the users (nodes) in order to explore the field of grid computing. Even now, this research problem remains obscured. In [4] a new load balancing algorithm for the grid computing service is proposed. This algorithm is based on the CPU speed of the workers in the grid. In [5] the author proposes new scheduler in which job allocation is determined by applying a Nash equilibrium solution. The system POPCORN [6] uses a single and double auction schema for job scheduling. In [7] authors show how to derive a unified framework for addressing network efficiency, fairness, utility maximization and pricing strategy for efficient job allocation in mobile grids. Their results show an asymptotically optimal behavior. In [8] Kwok et al. present hierarchical game-theoretic model of the grid, while they focus on the impact of non-cooperation in intra-site job execution mechanisms. Using a novel utility function they derive the Nash equilibrium and optimal strategies.

While investigating the non-altruistic behavior of the users in the grid computing, these papers do not explore the case when the benefit the users get from the grid is for the common good. We show that this approach must introduce some instantaneous benefit for the users in order to obtain and preserve their cooperation for achieving the global benefit. Thus, the contribution of this paper is twofold: we investigate the grid behavior in order to attract a user; and how it adapts in order to preserve the attracted node.

3. Node-Grid Game

We study grid computing environments that offer some processing involvement to a node but for a certain price. When a node tries to participate in the grid environment, it gains remuneration and certain obligations, such as fixed cost time for processing jobs for other nodes. In order to participate, the node must have higher benefit than loss. This is modeled using game theory.

The players in the game are the node and the grid environment. The node wishes to lessen its time spent on certain job execution. The aim of the grid is to ensure that the node will involve itself in the grid environment. The grid environment will be improved if the node participates more (i.e. gives more resources or pays higher price).

When the node has a job for execution it has two strategies: it can operate autonomously and run the job by itself, or it can participate in the grid environment and distribute the job to the nodes in the grid. When it executes the job unaided, the node has no obligations to the other nodes. However, if it joins the grid community it is expected to fulfill the grids requirements and participate in executing someone else's jobs.

Today, the CPU load of an average computer user, using everyday applications, is most of the time very low, this means that the CPU is overloaded at certain infrequent intervals. Thus, it is very important for the grid, i.e. the SETI@Home project to attract more nodes since it will gain more benefit while the node will ask less from the grid. When the grid negotiates with a new node it requires a certain amount of its resources (i.e. requires certain percentage of CPU cycles or memory storage) in exchange for the parallel execution of the new node's job. The node can either accept or reject the grid requirement. In this way we present the interaction between the node and the grid in an abstract level.

4. Players

4.1. Node

Each node has some computing power expressed as number of instructions executed in a time unit. We denote these capabilities as c MIPS (Million Instructions per Second). We assume that the node has a certain job to execute. The job size is m (Million Instructions) with data d that needs to be processed. The data size d depends on the number of instructions the job has, i.e. $d=D(m)$. The standalone time the node spends on execution of the job is T_{alone},

$$T_{alone} = m/c \tag{1}$$

At the beginning of the game the node announces the number of instructions of the job m, and the fraction of its executing capabilities it can offer to the grid, f. Depending on the computation power offered by the grid, the node can choose either to execute the job alone or to cooperate in the grid. The utility function of the node consists of the time it spends for executing the job. The node will participate if the job execution time spent in the grid is less or equal to the time needed when the node is not a member of the grid.

4.2. Grid

We assume that the grid environment consists of n homogenous nodes, the jobs are homogenous and that the grid offers fraction g of its overall computation power to the demanding new node. We also take into account the delay caused by the data flow on the network. The main aim of the grid is to try to get the node to participate in the grid. The node participation is crucial to the grid, because it increases its computation power, which gives good reputation to that particular grid and also by that the node helps in achieving some global worthy goal like cure for cancer, earthquake simulations, climate and financial modelling.

Focusing on the time constraint, the participation level of the node can be measured with the time it spends on job execution. Hence, the utility function of the grid is

$$T_{GRID} \sim n * f * c \tag{2}$$

The benefit of the grid is presented as an increasing function of the number of the nodes that participate and the fraction of the CPU cycles the nodes are willing to offer.

5. The Game – Attracting a Node

We study a game with an honest node, which either executes its jobs autonomously or participates in the grid and contributes to the grid by executing other nodes' jobs. The first aim of the grid is to make the node believe that it is better for it to be cooperative and to participate in the grid. If the node decides to execute the job by itself then the time it spends is given in (1). If the node participates in the grid, the time it spends is:

$$T_{coop} = T_{grid} + T_{net} \tag{3}$$

where T_{grid} is the time needed for the grid to execute the job and T_{net} is the time needed for the node to transfer the data to the destination nodes in the grid and to get the results back. We model the time constant T_{net} as:

$$T_{net} = \frac{f(m)}{NET} \tag{4}$$

where the network throughput (in Mbps) is presented with the NET constant.

Because most of the scientific problems are represented by matrix multiplication we choose the function $D(m)$ to correspond to a matrix calculation problem [9]. Thus, the data that will be sent through the network is:

$$d = \sqrt[3]{m^2} \tag{5}$$

From (3), (4) and (5) we have:

$$T_{coop} = \frac{m}{g*n*c*f} + \frac{\sqrt[3]{m^2}}{NET} \tag{6}$$

where g is the fraction of the number of nodes n that the grid is willing to offer for executing the job from the node.

The job has m instructions and the nodes offer fraction f of their computing power c.

We model this game as an extensive game and the structure of the game is as follows:

1. The node announces its job with m instructions.
2. The grid offers fraction g of its computing power.
3. The node either accepts or rejects the offer to become a member of the grid.

The optimal strategy for the node is evident. The node will participate in the grid if $T_{coop} \le T_{alone}$, otherwise it will remain a standalone node and will execute the job by itself. Hence, the utility function of the node is:

$$T = \min\left(\frac{m}{g*n*c*f} + \frac{\sqrt[3]{m^2}}{Net}, \frac{m}{c}\right) \tag{7}$$

If the grid offers fraction of its nodes where T_{alone} is smaller than T_{coop}, then the node operates unaided and the grid gains nothing, hence the utility of the grid is:

$$U_{grid} = \begin{cases} 0, & if\ T_{alone} < T_{coop} \\ T_{GRID}, & if\ T_{alone} \ge T_{coop} \end{cases} \tag{8}$$

The strategy space of the grid would be to offer contribution in the range [g, 1] and because the grid is a rational player it would offer fraction g of its computing power. This is Nash equilibrium of the game because the node and the grid can not increase their benefit if they unilaterally change their strategies. It is also a pareto-optimal or social optimal case because the sum of the benefit of both players is the highest.

From (6) and (7) we obtain that the grid should offer contribution:

$$g = \frac{NET}{n*(f*Net - c*f*\sqrt[3]{m^{-1}})} \tag{9}$$

In order to analyze the influence of the network throughput on the fraction the grid will offer to a node in order to attract it, we made several simulations. We observe a network with 100 nodes (n=100) and we assume that the node has a matrix multiplication job of size k=10^5. Because the type of the job is matrix multiplication, the number of instructions that will be executed is equal to m=k^2=1x10^4 million instructions [9]. The grid is homogeneous and is made up of nodes which have a CPU with c=10^5 MIPS (i.e. very close to the peak performance of the Pentium 4 Extreme Edition [10]), and the fraction each node gives is f=0.1.

On Fig. 1 the fraction g the grid must offer in order to attract the node depending on the network throughput is presented. The y-axis presents the speedup defined as a ration between the time spent when executing the job on its own and the time spent when executing the job in the grid. The node will join the grid only if the speedup is larger than 1. It can be seen that for a network with 5Mbps throughput or lower the node will not participate even if the grid offers 100% of its resources. If the network has 6Mbps throughput then the grid must offer about 45% of its resources. For bigger bandwidths the fraction that the grid must offer stabilizes around 10% of its resources.

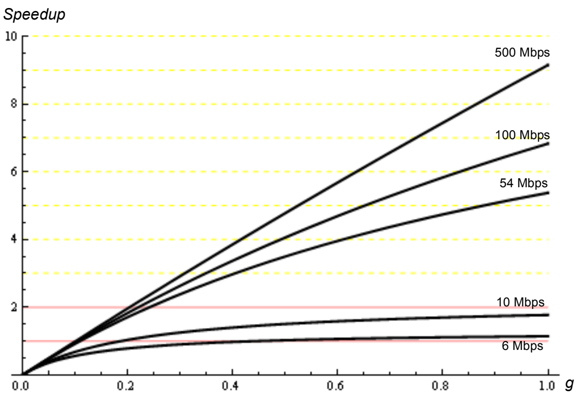

Fig. 1 Job's speedup depending on g for different network throughputs

In real life situations, it is not enough for the grid to offer a portion of its resources that makes the execution time of the job in the grid the same as the time the node needs to execute the job on its own. The nodes will not join the grid unless they gain speedup that is at least l times (l > 1), that is the job will be executed l times faster if the node is part of the grid.

From Fig. 1, it can be concluded that for l = 2 and network throughput of 10 Mbps or lower the grid cannot attract the node. The lowest throughput for attracting node that wants speedup of 2 is when NET=12Mbps and then the grid must offer around 90% of its resources; when NET=54Mbps it must offer around 30% of its resources; when NET=100Mbps it must offer around 25% of its resources and when NET=500Mbps it must offer around 20% of its resources.

6. The Game – Preserving a Node

When the grid succeeds in attracting the node, half of the work is done. The next step for the grid is to form the contribution (the fraction of grid computing power) it must offer to the node in order to preserve the node in the grid. It is obvious that the grid will choose this value depending on the amount of node contribution (fraction of CPU cycles) in the grid. If the rate of jobs that need to be executed by the node itself (this can be CPU bound jobs or jobs where security is an important aspect) is δ and the rate of jobs that can be executed in the grid is λ, the time needed for a node to execute these δ+λ jobs with m instructions is:

$$T_{alone} = (\delta + \lambda)\frac{m}{c} \tag{10}$$

This is the time the node needs to execute the jobs when it does not participate in the grid. When the node participates in the grid the time needed to finish its jobs will be:

$$T_{coop} = \delta\frac{m}{(1-f)*c} + \lambda\frac{m}{g*n*c} + \frac{\sqrt[3]{m^2}}{Net} \tag{11}$$

Based on (10) and (11) we show how the grid must react to a given contribution of a node in order to preserve the node in the grid by offering minimal fraction g given by:

$$g = \frac{Net * (1-f)}{n * (\sqrt[3]{m^{-1}} * c(f-1) + m * Net(1-f-f*\rho))} \quad (12)$$

where:

$$\rho = \delta / \lambda \quad (13)$$

where ρ represents the ratio between the rate of the CPU bound jobs that must be executed by the node itself and the rate of the jobs that can be executed in the grid. From (12) we can calculate the maximum ratio for which the grid can still preserve the node in the grid:

$$\rho = \frac{(c - \sqrt[3]{m} * Net) * (f-1)}{f * \sqrt[3]{m} * Net} \quad (14)$$

In order to analyze the influence of the ratio ρ on the contribution that the grid must offer in order to preserve the node, we investigate a scenario where the average instructions per job of a node are m=1x10^4 Million Instructions, the capacity of the CPUs in the grid is c=1x10^5 MIPS, and where the nodes offer 20% percent of its processing power for the grid (i.e. f=0.2). The ratio ρ is between 0.1 and 5, where 0.1 means that there are 10 times more jobs that can be executed in the grid, while 5 means that there are 5 times more jobs that the node must execute alone. The number of nodes in the grid is between 1 and 100.

On Fig. 2 the network throughput is 100 Mbps. It can be seen that as the ratio ρ grows, the grid must offer more of its computing power in order to preserve the node. Also as the number of nodes in the grid is increasing, the grid lessens its contribution. There is a certain preservation limit, i.e. when the ratio ρ is larger than its maximum in (14) in which case there should be some incentive mechanism if the grid wants to keep the node. In the observed example the ratio cannot be larger than 4, however this value largely depends on the chosen parameters.

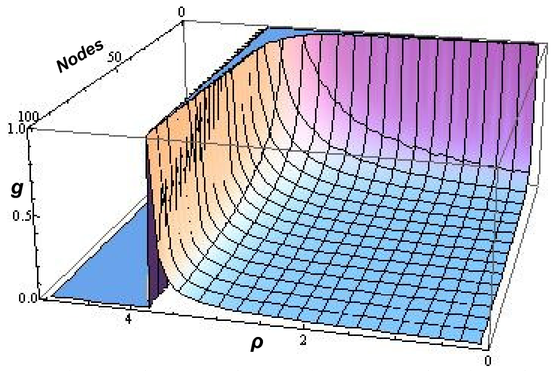

Fig. 2 Influence of the ratio ρ on the fraction of computing power offered by the grid

7. Conclusion

In this paper, we address the problem of cooperation between a node and a grid system. We assume that users are rational and we show that they are willing to cooperate only if the grid offers some instantaneous contribution to them. By using elementary game theory we were able to show the existence of an operating point wherein the grid is able to attract and preserve the nodes with mutual benefit. Then we show that the participation of the node also depends on the ratio of CPU bound jobs that node must executed by itself and jobs that can be executed in the grid.

We would like to highlight that the aim of this work is to provide a mathematical framework for studying user participation in computational grids. Further research will be required in order to force the nodes to cooperate by some payment and/or punishing mechanisms. We also want to investigate how the probability for a next round will affect the user's actions.

References

[1] Luiz A. DaSilva and Vivek Srivastava, "Node Participation in Ad Hoc and Peer-to-Peer Networks: A Game-Theoretic Formulation," *Workshop on Games and Emergent Behavior in Distributed Computing Environments,* September, 2004, Birmingham , U.K.

[2] "Search for Extraterrestrial Intelligence at Home," 2003; http://setiathome.ssl.berkeley.edu/

[3] D. Fundenberg and J. Tirole, "Game Theory," *MIT Press*, Cambridge, Massachusetts, 1991.

[4] Abderezak Touzene, Hussein Al Maqbali, "Performance of Load Balancing for Grid Computing," *Proc. of the 25th IASTED International Multi-Conference Parallel and Distributed Computing and Networks*, Austria, February 2007.

[5] Massimo Orazio Spata, "A Nash-Equilibrium based Algorithm for Scheduling Jobs on a Grid Cluster," *Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2007, pp. 251-252.

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

[7] Preetam Ghosh, Nirmalya Roy, Sajal K. Das and Kalyan Basu, "A Pricing Strategy for Job Allocation in Mobile Grids using a Non-cooperative Bargaining Theory Framework," *Journal of Parallel and Distributed Computing*, 2005.

[8] Yu-Kwong Kwok, ShanShan Song, and Kai Hwang, "Selfish Grid Computing: Game-Theoretic Modeling and NAS Performance Results," *Proc. CCGrid 2005*, Cardiff, UK, May 9–12, pp.1143–1150.

[9] Rohit Chandra: "Parallel Programming in OpenMP," *Elsevier Science & Technology Books*, October 2000.

[10] "Intel Pentium 4 3.2GHz Extreme Edition Processor Review",http://www.pcstats.com/articleview.cfm?articleid=808&page=6