Using Game Theory to Analyze Distributed Computing Systems

Igor Mishkovski, Sonja Filiposka, Dimitar Trajanov, Aksenti Grnarov and Ljupco Kocarev, Member, IEEE

Abstract — In this paper we use game theory to study node behavior in distributed systems. Single stage game of complete information and infinitely repeated game is used to give prescription of node’s behavior. In a single stage game nodes will be non-cooperative, but when the game is infinitely repeated their strategy depends on the discount factor, i.e. the probability for a next round. Using game theory we also model the interaction between a node and the distributed environment as a whole.

Keywords — distributed computing, game theory, grid computing, nash equilibrium.

I. INTRODUCTION

The need of distributed resources is greater than ever and become more practical with the development of the Internet. The Internet and its current state offers big possibilities, which if they are wisely used they can be of a great importance and relevance of the human kind. As a result there are a wide variety of implemented distributed systems [1]. These distributed systems can offer users computer resources in order to speed up the execution time of a certain complex and time-dependent computer program by distributing it for execution across the CPU’s in the network. Nodes that can accomplish their tasks fast can receive some payment from the network authority. Having this in mind the forthcoming question here is whether for the users (i.e. nodes) it is better to cooperate and give some portion of the program to be executed by the other users.

In order to analyze the distributed systems we propose game theory as a mathematical tool [2].

Game theory introduces mathematical background for different analysis of the interactive processes for decision making. This theory enables tools that can leverage the prediction of what might happen in an environment in which there is interaction between agents with conflict interests, i.e. non-cooperative environment. The traditional applications of Game Theory tries to find out the equilibrium point, i.e. set of strategies in which is almost impossible for the individuals to change the current strategy.

This theory became was introduced in [3] and its further development was due to the Nash Equilibrium concept in [4]. The games that were studied during the evolution of this theory were well defined mathematical objects. The games are consisted of players, a set of strategies, and specification of the profits for every combination of the strategies.

In this study we analyze the case with two nodes in a single stage static game of complete information and we have found out that for a single stage game the strategy that nodes choose in our model, is not to cooperate, i.e. they execute their task by themselves instead of cooperating and by that they share its tasks with the other nodes. Additionally we extend the model by introducing a discount factor [5] in an infinitely repeated game. We calculate the value that this discount factor must satisfy in order a Trigger strategy [2] to be Nash equilibrium. Additionally we introduce another concept where we try to model the relations between one node and the distributed environment as a whole.

The rest of the paper is as follows. In Section 2 we present an introduction to Game Theory an in Section 3 the related work in using a game theory as mathematical model for nodes involvement in distributed systems is presented. Section 4 depicts the single stage game for node participation in distributed systems, while Section 5 extends the game to infinitely repeated game. Section 6 models the interaction between a node and the distributed environment as a whole. Section 7 concludes and presents future work in this field.

II. RELATED WORK

In recent years Game Theory makes big steps in analysis of the distributed computer systems. In [6] 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 [7] 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. In [8] authors, using Game Theory,
derive the conditions under which users will participate in non-commercial Grid projects. In [9] Game Theory is used as a tool to compare and analyze Grid resource allocation mechanisms. The nodes behavior in peer-to-peer networks when nodes receive a service based on their reputation is studied in [10].

Our contribution is: 1) we use infinitely repeated game to describe the user’s participation in distributed systems; 2) we propose various game theoretical analyses that fit well in distributed systems.

III. SINGLE STAGE STATIC GAME OF COMPLETE INFORMATION BETWEEN TWO NODES

We can define a game \( G = (N, S, U) \) in normal form with the following three elements: \( N \) – set of players; \( S_i \) – pure strategy space of the player \( i \). The pure strategy assigns probability 0 to all the moves, except one, i.e. it clearly defines what action the player should play. In the case of mixed strategies, the players play the actions with a certain probability. The probability distribution of the pure strategies for one player, \( s \in S_i \), is denoted as \( \sigma_i(s) \). The joint set of the strategy space of all players is denoted as \( S = S_1 \times S_2 \times ... x S_N \). With \( S_i = S \setminus S_i \) we denote the pure strategy space of the opponents of the player \( i \). The set of the chosen strategies forms the strategy profile \( s = \{s_1, s_2 \ldots \} \). The profit or the utility \( u_i(s) \) denotes the outcome of the player \( i \) with a given profile of strategies \( s \). For example, for two players we would have \( U = \{u_1(s), u_2(s)\} \).

One solution concept of a game involving two or more players is the Nash equilibrium [6]. This concept relies on the best response of player given the possible strategies of the other players. Thus, the best response of the player \( i \) on the strategy profile \( s_i \) is the strategy \( s_i^* \):  

\[
br_i(s_i) = \arg \max_{s_i \in S_i} u_i(s_i, s_{-i})
\]  

(1)

It is obvious that if the two strategies are mutually the best responses, then no one would have a motive to alter his strategy so it could harm the others. Thus, we can define the concept of Nash equilibrium: A profile of pure strategies \( s^* \) is Nash equilibrium, if for every player holds:  

\[
u_i(s^*_i, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \ \forall s_i \in S_i
\]  

(2)

This means that in the state of Nash equilibrium, no one would unilaterally change his strategy and increase his profit.

Using game theory and the Nash equilibrium concept our aim is to model the nodes behavior in distributed computing systems, i.e. Grid computing systems. The problem in these systems is that users are not always willing to cooperate and there must be some incentive mechanism or central authority that will make some payment to the nodes that are willing to cooperate.

We model node participation in distributed computing as a strategic-form game. The players in the game are the nodes and both players have the same strategies: to take part in the game; or to be non-cooperative and execute their tasks by their own. The both strategies are known to each other. We represent the tasks that the nodes have with serial portion and a parallel portion. The time needed for a node to execute the serial portion is \( a \) and the time for the parallel portion is \( b \). We investigate the scenario with two players which have tasks \( task_1 \) and \( task_2 \), respectively, ready to be executed. For simplicity we assume that both nodes have tasks with same size and their parallel and serial portions are equal (i.e. \( a_1=a_2=a \) and \( b_1=b_2=b \)). In the game there exists an incentive mechanism (i.e. central authority) that awards the players if he finishes its task quickly, represent by a constant \( m \). If both players defect they will execute only their tasks. If player 1 cooperates and player 2 defects then player 1 must execute his task and half of the parallel portion of the task of the player 2. Thus reducing the time player 2 spends for execution of its task and by that player 2 receives award \( m \). The payoff matrix in table 1 models the benefits and the costs of the game.

### Table 1: Normal strategic form of the game

<table>
<thead>
<tr>
<th>PLAYER 1</th>
<th>( \text{Defect} )</th>
<th>( \text{Cooperate} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Defect} )</td>
<td>(-a+b/2) + m</td>
<td>(-a+b) + m</td>
</tr>
<tr>
<td>( \text{Cooperate} )</td>
<td>(-a+b/2) + 2m</td>
<td>(-a+b) + 2m + m</td>
</tr>
</tbody>
</table>

By using (1) and (2) it is evident that the Nash equilibrium for the players is \( (D,D) \). This means that for a single stage static game it is better for the nodes not to cooperate and this may lead to a suboptimal social equilibrium depending on the award from the incentive mechanism.

IV. INFINITELY REPEATED GAME BETWEEN TWO NODES

We suppose that the previous game is to be repeated infinitely and that for each \( t \), the outcomes of the \( t-1 \) previous plays are observed before the \( t \)th stage begins. We use \( \delta \), a discount factor to interpret an infinitely repeated game as a repeated game that ends after a random number of repetitions. In one word this discount factor represents the possibility that the game will end. We will show that cooperation – that is, \( \text{(Cooperate, Cooperate)} \) – can occur in every stage of a subgame-perfect outcome of the infinitely repeated game, even though we showed that the only Nash equilibrium in the stage game is non-cooperation (Defect, Defect). If we assume that player 1’s strategy is trigger strategy [2], that is: play cooperative in the first stage. In the \( t \)th stage, if the outcome of all \( t-1 \) preceding stages has been (Cooperate, Cooperate) then play Cooperative; otherwise, play Defective.

We want to show what the outcome will be if both players adopt the trigger strategy. We assume that player 1 adopted the trigger strategy and we want to show when it is best for player 2 to adopt this strategy. If player 2 chooses to make the best response to the strategy of the player 1 he will choose not to cooperate but this will force player 1 not to cooperate also in the further states of the game. So the payoff the player 2 gets by this is:  

\[
V = -(a+b) + m + \delta V
\]

(3)

Alternatively, playing cooperative will yield a payoff of:  

\[
V = -(a+b) + m + \delta V
\]

or:
is the time the node spends. We assume that the
\[ \delta \geq \frac{1}{1+2m/b} \]  
(6)

Thus, if all preceding outcomes have been (Cooperative, Cooperative), player 2’s optimal strategy (given that player 1’s strategy is trigger) is to cooperate, if and only if the inequality is satisfied. The conclusion is that if both players adopt this strategy then the outcome of the infinitely repeated game might be cooperation in every stage. This cooperation depends on the discounting factor.

In fig. 1 the shaded area represents the values of the discount factor when this trigger strategy is Nash equilibrium. The x-axis represents the proportion between the award that the central authority gives to a collaborating node and the time a node spends for executing the parallel part.

From the figure we can conclude that if the parallel part, \( b=1 \) and the award is below 0.2 then the probability for the next stage of the game must be high enough, i.e. the probability must be more then 0.7. The increasing of the award drops the probability for a next round drops. We can see that if reward is twice as bigger than the time spent on executing the parallel part then the probability for the next round can be as least as 0.2.

Fig. 1. Values of the discount factor when trigger strategy is NE, depending on the ratio \( b/m \)

V. EXTENSIVE GAME OF COMPLETE INFORMATION BETWEEN A NODE AND THE DISTRIBUTED ENVIRONMENT

In this part, instead of modeling the relations between two nodes, we try to model the relations between one node and well established distributed system, i.e. grid computing environment. Thus, we study distributed systems that offer some processing involvement to a node but for a certain price.

When a node tries to take part in the grid environment, it gains remuneration and certain obligations, like fixed cost time for processing tasks for other nodes. In order to take part, the node must have bigger benefit than loss.

The players in the game are the node and the grid environment. The node wishes to lessen its time spent on certain task execution. The aim of the network 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).

If the node has some task for execution it has two possibilities. The node can either operate autonomously or run the task by itself or it can participate in the grid environment and distribute its task to the nodes in the grid for execution. In the case when it executes the task unaided it has no obligations to the other nodes. If it joins the grid community it is expected to participate in executing some other tasks. The tasks the node may have are with the same execution time and they only differ by how much they can be parallelized. It is known that not everything can be parallelized so the node divides the task in a part which may be run on the grid and some other part that must be executed by the node itself:

\[ T(\sigma) = a(\sigma) + b(\sigma) \]  
(7)

where \( T(\sigma) \) is the time needed for executing a certain type of task \( \sigma \). Depending on the type of the task, it can be divided into a part that must be executed by the node \( a(\sigma) \) and another part that can be parallelized \( b(\sigma) \). At the beginning of the game the node announces its type of task. Depending on the price offer of the environment, the node can choose either to be alone or to cooperate. The utility function of the node consists of the time it spends for the execution of the task. The difference between the two alternatives of the node is the time it spends during the execution of the task, thus the utility function is

\[ T_{no} = -t \]  
(8)

where \( t \) is the time the node spends. We assume that the node participates in the grid if the time it spends if it cooperates is smaller or identical with the time it spends when it executes the task alone.

When a node wants to participate, the grid requires a certain involvement in the grid functions in exchange for the execution of the task. The node then has two possibilities: either to accept or to reject the grid requirement. In this way we present the interaction between the node and the grid in an abstract level.

The second player, the grid environment, consists of \( N \) homogenous nodes with same computational characteristics. We also neglect the influence of the network in the final price of the grid environment and we assume that there is no mobility in the network. The main aim of the grid is to try to get the node to participate in the grid, since participation of the node reduces the overall time spent. The participation of the node is crucial to the grid, because it increases its computational capacity, which gives good reputation to that particular grid.

Because we focus on the time constraint, the participation level of the node can be measured with the time it spends on execution tasks. Hence, the utility function of the grid is:

\[ T_{no} = c \]  
(9)

where \( c \) is the time that the node spends on executing grid tasks.

Here we study a game with an honest node, which either executes its tasks autonomously or participates in the grid and contributes to the grid by executing other nodes’ tasks. If the node decides to execute the task by itself then the time it spends for executing a certain task is
\( T(\sigma) = a(\sigma) + b(\sigma) \). If the node participates in the grid the time it spends is:

\[
\frac{T(\sigma) - a(\sigma)}{N} + a(\sigma)
\]  

(10)

If the node uses the nodes in the grid then it should also contribute in task execution of the other nodes. The involvement of the node requires contribution \( c \).

We model this game as an extensive game and the structure of the game is as follows: the grid environment offers to execute the task in exchange for computation effort \( c \); then the node either accepts or rejects the offer.

The optimal strategy for the node is evident. If \( c \leq c_0 = (T(\sigma) - a(\sigma)) / N - a(\sigma) \), the node uses the grid for task execution, otherwise it executes the task by itself which results in utility:

\[
U_{N0} = \max((-T(\sigma) - a(\sigma)) / N - a(\delta) - C, -T(\delta))
\]  

(11)

If the grid offers contribution \( c \) greater than \( c_0 \), the node operates unaided and the network gains nothing, hence the utility of the network is

\[
U_{Ne} = \begin{cases} 
0, & \text{if } c > c_0 \\
\delta, & \text{if } c \leq c_0
\end{cases}
\]  

(12)

The optimal strategy of the grid is to require contribution \( c_0 \). The solution is that grid requires contribution \( c_0 \) and the node participates in the network. This is Nash equilibrium of the game and it is also a pareto-optimal or social optimal case. The forthcoming question is how the grid knows which price it should offer to a certain node. The answer lies in the type of the task that one node has. Hence the grid environment can offer different prices for different types of tasks. These prices can be deducted by giving the grid certain learning capabilities.

A. Example

Suppose that the grid network is consisted of 100 nodes and in moment \( k \), node \( i \) has a certain task \( \sigma \) for execution. This task can be partitioned in some parallel part \( b \) and serial part \( a \). After knowing the task, the mission of the grid is to offer a price which satisfies both the node and the grid itself. Without any previous knowledge this could be tricky. But if the grid has learning ability it can offer the best price which also satisfies the node. In Table 1 different NE prices offered by the grid are given depending on the type of the task. We assume that the execution of the whole task requires one time unit.

<table>
<thead>
<tr>
<th>Task type</th>
<th>Serial part</th>
<th>Parallel part</th>
<th>Offered price</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_0 )</td>
<td>0.1</td>
<td>0.9</td>
<td>0.109</td>
</tr>
<tr>
<td>( \sigma_1 )</td>
<td>0.5</td>
<td>0.5</td>
<td>0.505</td>
</tr>
<tr>
<td>( \sigma_2 )</td>
<td>0.9</td>
<td>0.1</td>
<td>0.901</td>
</tr>
</tbody>
</table>

This different types of tasks can represent different types of real life problems like matrix calculation, PI calculation etc. The same NE offered prices can be shown in fig. 2 where the vertical axis represents the cost that a node has for executing its task. The green surface represents the time lost when the node executes the task by itself. The blue surface represents the time the node will lost when it cooperates and execute its task in the grid. Normally this surface depends on how much the task can be parallelized and how much contribution \( c \) would the grid environment require from the node.

Fig. 2 Node’s cost depending on parameters \( k \) and \( c \)

VI. Conclusion

In this paper, we have analyzed nodes behavior in distributed systems by proposing single stage static game of complete information, where the players in the game are the nodes in distributed systems. In this game we proved that the Nash equilibrium of the game is for player not to cooperate. Additionally we extend the game and showed that if the game is infinite and that there exists probability for a next round the users might cooperate like in the Prisoner’s dilemma. We also used different kind of approach in which the players are the node and the distributed environment as a whole and show when for the node is better to cooperate and how much the environment must offer in order to attract the node, depending on the type of the task the node has.

In future, we want to develop models for \( N \) heterogeneous players and these models besides the type of the job will also include the rate of job creation among nodes, the network environment (network path loss exponent, link bandwidth, transmission range of the nodes, etc.).

REFERENCES