Malicious Workers Tolerance in an Agent-Based Grid Resource Brokering System – Preliminary Considerations

(Short Paper)

Naoual Attaoui
Information and Telecommunication Systems Laboratory
Faculty of Science, Tetuan, Morocco
Email: naoual_attaoui@hotmail.com

Mohammad Essaaidi
National School of Computer Science and Systems Analysis
ENSIAS, Rabat, Morocco
Email: essaaidi@gmail.com

Marcin Paprzycki (ca), Maria Ganzha
Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland
Email: firstname.lastname@ibspan.waw.pl

***Abstract*—The Agent in Grid (*AiG*) project develops an agent-based infrastructure for resource management in the Grid. The basic assumptions of the project are that (1) agents work in teams, and (2) information is ontologically represented and semantically processed. Thus far, issues involved in trust management were considered only from the point of view of fulfillment of contracts between stakeholders. However, it is also possible that some *workers* that fulfill their contracts, return incorrect results. In this note, we consider how the trust management in the *AiG* project can be further conceptualized by using a reputation-based voting.**

Keywords: Grid computing, agent system, trust management, malicious workers tolerance, majority voting, reputation

I. Introduction

Grid computing allows, among others, use distributed heterogeneous resources to complete large-scale computations, which may not be easily realizable otherwise. Computational Grids can be considered in two "scenarios." First, when resources forming the Grid are bound by formal agreements and administered by appointed administrators (i.e. in *closed* or *desktop* Grids). Here, trust management is not "necessary," as all issues are governed by formal contracts between participants, who formed the Grid and agreed to obey the rules. Second, the Grid is an *open environment* where resources can be bought and sold. Here, due to the openness towards the world, users (resources) of *unspecified* origin can participate in computations. As a result, it is possible that *malicious* participants provide false job results to the users. Therefore, the (global) Grid should have mechanisms to defend itself against malicious *workers*. Only then, the trust of the users, especially from the business community, can be build (see [Ermisch J. and Gambetta D., 2006]). These mechanisms have to guarantee authenticity, confidentiality and integrity of results.

In this context, an agent-based Grid resource management system was proposed (the Agents in Grid (*AiG*) project). However, in the *AiG* system, trust management was considered only from the perspective of contractual relations between stakeholders. At the same time, the possibility of existence of malicious *workers* was not considered. Nevertheless, it is clear that mechanism establishing correctness of results (assurances of validity of job results) have to be introduced (see, also [Kumar P.S. et al., 2011]). Preliminary discussion of one possible approach to achieving this goal is the aim of this note.

To this effect, in the next section, we briefly summarize the related work. We follow with an overview of the *AiG* system, including trust management issues. Finally, we propose a solution to deal with malicious resources.

II. Related work

Reputation systems [Resnick P. et al., 2000] are commonly used to estimate the reliability of Grid resources. They are based on the history of correctness of results delivered by the *workers*. In these systems, stakeholders share information about the trustworthiness of others. The advantage of this approach is in its simplicity, while its disadvantage is that it has problems in dealing with *workers* that behave well for a long period of time, in order to gain credibility, and after that start to sabotage the results. Furthermore, collusion of a group of *workers* is rather difficult to handle (see [Hoffman K. et al., 2009], [Luke Teacy W. T. et al., 2006]).

A replication-based mechanism, also known as majority voting, is used for ensuring correctness of results (to detect and tolerate erroneous results) in volunteer systems such as BOINC, SETI@home, Folding@home and Mersenne (see [BOINC website], [Cuenca-Acuna F. M. et al., 2003]). It is based on replicating each task to multiple *workers*. Next, the returned results are verified by the master, using a simple voting technique. The main benefit of this approach is its simplicity, while its major weakness lies in wasting resources that are used to complete the same task multiple times.

Credibility-based approach was proposed in [Sarmenta L. F. G., 2002]. It is based on combining voting and spot-checking (thus balancing loss in performance and desired correctness). Here, in order to check the credibility of *workers*, the master assigns a spotters task (with a certain probability q) and applies voting to the obtained results. This approach can mathematically guarantee that the error rate will not exceed a given (acceptable) threshold ϵ_{acc}.

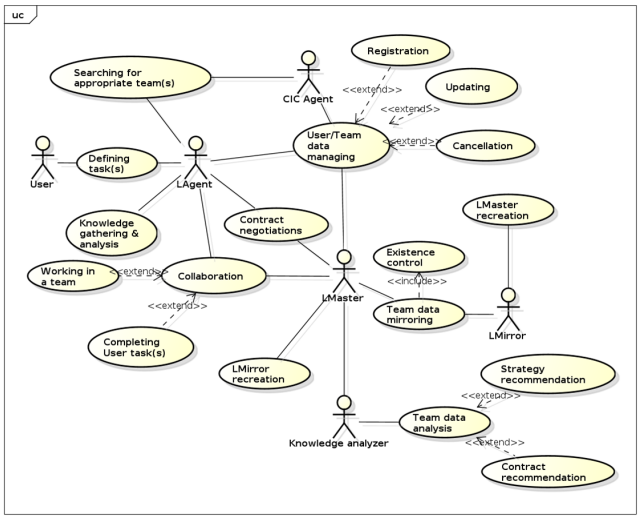

Fig. 1: Use Case Diagram of the *AiG* system.

A reputation-based voting technique was recently proposed in [Bendahmane A. et al., 2010]. It is an improvement over the credibility-based approach as, in addition to the majority voting, it *also* involves reputation. Therefore we have decided to check how this solution could be applied in the *AiG* system.

III. The *Agents in Grid* project – overview

The *AiG* project attempts at following the ideas originally outlined in [Foster I. et al., 2004] and combine strengths of Grid and agent approaches. The project uses ontologies to represent knowledge and semantic data processing to take advantage of it. In the system, agents work in teams (see [Kuranowski W. et al., 2008a], [Kuranowski W. et al., 2008b]). This is to mitigate resource disappearance that has to be taken into account in an open Grid. Each team is managed by its "leader," the *LMaster* agent. Agent teams utilize services of the *CIC* agent, to advertise work they are ready to do, and skills of *workers* they would like to hire [Dominiak M. et al., 2006]. Each team includes an *LMirror* agent, which stores a copy of the information necessary for the team to persist if the *LMaster* crashes. The proposed approach can be represented in the form of a Use Case Diagram depicted in Figure 1. Work of the system can be visualized through two scenarios: (1) *User* that wants to execute a task (using Grid resources), and (2) *User* that wants to join a team (to sell resources and earn money). When the *User* is seeking a team to execute its job, it specifies job execution constraints to the *LAgent*. The *LAgent* contacts the *CIC* to obtain the list of teams that can execute the job and utilizes trust information to select teams that can be trusted to do the job "right." Next, negotiations between the *LAgent* and the *LMasters* representing selected teams ensue and, hopefully, result in an agreement. The team-joining scenario follows the same pattern; the only difference is the information that is passed around and the content of the reached agreement.

IV. Trust management in the *AiG* system

Obviously, trust is important for Grid computing. Here, users must trust that the providers will deliver the requested resource, while the provider must trust the user will pay for the services. Therefore, in the *AiG* project, we have initially considered basic issues involved in trust management. As discussed in [Ganzha M. et al., 2007], we have identified four situations that are influenced by the trust between: (1) *Users* and *LMaster*s (teams that they represent), and (2) *LMaster*s and their *Workers*:

1) In the team-joining scenario the *LAgent* obtains the list of teams that it can join and may want to consider only those that are deemed "trustworthy enough."
2) In the same scenario, when the *LMaster* receives a *Call-For-Proposal* from an *LAgent* that wants to join its team, it may want to reject proposals from *workers* that are deemed not "trustworthy enough."
3) In the job-execution scenario, the *LAgent* obtains the list of teams that can execute its task, and may want to send the *Call-For-Proposal* only to those that are deemed "trustworthy enough."
4) In the same scenario, when an *LMaster* receives a *Call-For-Proposal* concerning job execution, it should check the "trust value" of the *LAgent* that send the *CFP* and act according to it.

In [Ganzha M. et al., 2007], trust management issues concerning these four scenarios has been elaborated and specific proposals for each one of them put forward. However, an open Grid system should also ensure the integrity of results of executed jobs. Hence, it is necessary to protect the users against "cheating" by Grid resources. In other words, an open Grid environments should detect and deal with malicious resources, which tamper with the computation and return corrupted results. To achieve this goal we will now consider the approach proposed in [Bendahmane A. et al., 2010].

V. Proposed solution

A. Feasibility

Let us start from describing the Grid model appropriate for application of the majority voting mechanism (in [Bendahmane A. et al., 2010]) and juxtaposing it with the *AiG* infrastructure. Sabotage tolerance techniques of the type we are interested in can applied in Grid systems that employ the master-worker computational model [Bendahmane A. et al., 2010]. These approaches require a server that can distribute work units to the Grid nodes. For this purpose, in [Bendahmane A. et al., 2010], the concept of a *Virtual Organization* (VO) was applied. In Grid computing, a VO typically refers to a dynamic set of individuals or institutions defined around a set of resource-sharing rules and conditions. Figure 2 shows the basic components of a Grid system of interest, which consists of N Virtual Organizations (VOs). Here, we depict a situation when a client submits a job to the Grid broker service (after formulation of a service level agreement). The broker facilitates access to the Grid resources by (a) discovering needed services and resources provided by VOs, (2) deploying the task, and (3) monitoring its execution. In this model, each VO in the Grid has its own VO manager, which searches for available services within its VO. When it finds a relevant service, it negotiates with the service provider to allow access to that service. Once the

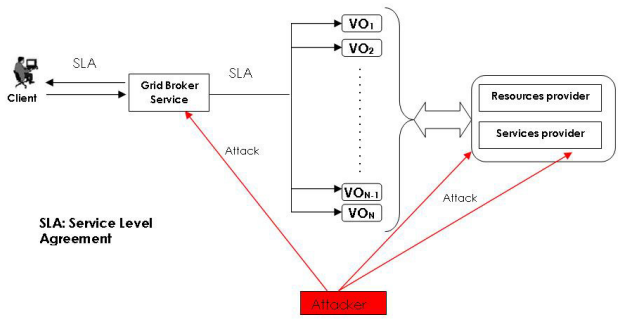

Fig. 2: The Grid model appropriate to the sabotage tolerance techniques.

service provider authorizes the use of the service, it becomes available to complete the job.

It should be immediately clear that this Grid model is very similar to that found in the *AiG* system. The main difference is that in the *AiG*, the *LAgent* negotiates directly with the *LMaster* of a team, without needing a Grid Broker service. This could be translated into direct negotiations with VO managers. In the *AiG* system, the Grid resources discovery is facilitated by the *CIC* Agent. To make these points clearer, the analogy between both models is depicted in Table I.

Since our long-term goal may be to implement the proposed result-trust management technique in the *AiG* system, let us see how the "reputation value" of each *Worker* can be established there. Following the proposal put forward in [Bendahmane A. et al., 2010], the reputation of each *Worker* should combine assessment of credibility, availability, and security level. Let us thus consider each one of them separately.

B. Credibility

The *credibility* can be computed by applying the spot-checking techniques from [Sarmenta L. F. G., 2002]. Here, the *LMaster* would randomly give a worker a spotter work, correct result of which is already known to it. This technique can include a Blacklist that allows exclusion of *Workers* that failed the tests. Note that in the *AiG* system, it would also be possible to store the Blacklist within the *CIC* infrastructure. Let us stress that the *CIC* has to be assumed assumed to be trusted for the *AiG* approach to work. Furthermore, it is assumed that each agent in the *AiG* system will have to register with the *CIC*. Therefore, the *CIC* is a very natural place to safely store the global Blacklist of untrustworthy *Workers*.

The credibility of a resource, which correctly computed K_i spotter tasks, would be computed using the following equation [Sarmenta L. F. G., 2002]:

$$CR(C_i, K_i) = \begin{cases} 1 - \frac{f}{1-f} \cdot \frac{1}{K_i \cdot e}, & K_i \neq 0 \\ 1 - f, & otherwise \end{cases} \quad (1)$$

Where f is the proportion of malicious workers that intentionally submit bad results, e is the base of the natural logarithm, and $1 - f$ is the minimum of credibility. The credibility value of the *Worker* that successfully completed a spotter task is incremented as follows (for details, see [Sarmenta L. F. G., 2002]):

$$CR(C_i, K_i) = CR(C_i, K_i + 1) \quad (2)$$

In the same manner, we decrement the credibility of those resources whose result was not validated by the reputation-based majority voting using formula:

$$CR(C_i, K_i) = CR(C_i, K_i - 1) \quad (3)$$

After resources pass enough tests (K_{min}), they succeed to obtain minimum credibility for the system to assume that their results are correct.

C. Availability

The *availability* A_i is the ratio of the number of successful contracts of *Worker* C_i, and the total number of requests. Here, a monitoring mechanism introduced in [Ganzha M. et al., 2007], can be used to check availability of workers. The availability value is calculated as follows:

$$A_i = \frac{N_S}{N_R} \quad (4)$$

Where N_S is the number of successful checks, and N_R is the total number of checks.

D. Security level

The *security level* of a computing resource was defined in [Chen C. et al., 2009]. It is calculated by aggregating the security factors like firewall, or anti-virus. Values of these factors belong to the interval $[0, 1]$. According to [Bendahmane A. et al., 2010], the security level is calculated using the following formula:

$$SL = \frac{\sum_{f=1}^{n} W(f)A(f)}{n} \quad (5)$$

Where n is the total number of factors, $W(f)$ is the weight of a factor, and $A(f)$ is the value of the factor.

E. Worker reputation

The reputation value R_i of a worker C_i is the product of credibility, availability, and security level, and it is computed using the following equation:

$$R_i = CR(C_i, K_i) \times A_i \times SL_i \quad (6)$$

F. Majority voting

Following [Bendahmane A. et al., 2010], the *LMaster* distributes n replicas of a task to several (selected) resources C_i , so that it can collect m different results V_j, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. Notice that $m \leq n$. Each collected result is seen as a vote in a voting pool with n voters. To decide, which result V_j is trustworthy (i. e. accepted result), the *LMaster* utilizes a majority voting based on the reputation criteria. Here, we assume that since all *LAgent*s have to be registered with the *CIC*, one could obtain their reputation values $R_i \in (0, 1)$ (from the *CIC*). Therefore, the *LMaster* can use the combined reputation score to decide if that worker is malicious or not. Let $T(V_j, C_i)$ represent the relationship

Model appropriate to Sabotage tolerance techniques	Model proposed by *AiG* system
VOs	Teams
VO manager	LMaster
Resources provider	LWorkers

TABLE I: Analogy between the Grid model appropriate for sabotage tolerance techniques and the *AiG* system

between the result V_j and the worker C_i. It is calculated by the *LMaster* as follows:

$$T(V_j, C_i) = \begin{cases} 1, & C_i obtained\ result\ V_j \\ 0, & otherwise \end{cases} \quad (7)$$

We define the resulting reputation $RR(V_j)$, of a given result V_j, as the sum of reputations of the workers returning the result V_j. Therefore, for each result V_j:

$$RR(V_j) = \sum_{i=0}^{n} T(V_j, C_i) \times R_i \quad (8)$$

Where R_i is the reputation of worker C_i. To make a decision about the most reliable worker(s) we fix a positive threshold value $\lambda < 1$ and we find the maximum of $RR(V_j)$; $(j = 1, ..., m)$. Here, λ depends on the trust level required by the team. If λ is high, the trust level of the Grid is going to be high. To avoid the possibility that a set of workers, with a low reputation, could undermine the result, we impose the following condition:

$$R_i = \begin{cases} 0, & if\ R_i < \theta \\ R_i, & otherwise \end{cases} \quad (9)$$

Where θ represents the minimum reputation value a *Worker* should have for its results to be taken into consideration. If the result produced by the reputation-based majority voting is accepted, the reputation of the worker will be increased. If not, it will be decreased (see, above).

Observe that the value of the reputation can be a good criterion of decisions to be made by the *LMaster*, which wants to choose the best worker to become its *LMirror* in case of its disappearance; or for the *LMirror*, that has to create a new *LMaster*.

We can also think that every *LMaster* can fix a minimum of reputation value (its own, private to the team). In this case, the incoming *Worker* that doesn't reach the threshold, will not be accepted to become a member of the team.

VI. Concluding remarks

The aim of this note was to discuss issues involved in *Worker* trust management, in an agent-based Grid resource management system. In an open Grid environment, it is extremely important to be able to deal with malicious workers that can purposefully corrupt the results of a job. To counteract this threat, in the *AiG* system, we have proposed to apply method recently introduced in [Bendahmane A. et al., 2010]. Taking into account the, overall positive, initial evaluation of the match between the *AiG* system and the proposed method, our next step will be to move towards its implementation.

References

[Bendahmane A. et al., 2010] *Reputation-Based Majority Voting For Malicious Grid Resources Tolerance*. Scalable Computing: Practice and Experience, Vol. 11, No. 4, pp. 385392.

[Ganzha M. et al., 2007] *Trust Management in an Agent-based Grid Resource Brokering System-Preliminary Considerations*. AIP Conference Proceedings, Vol. 946, No. 35, pp. 35-46.

[Kumar P.S. et al., 2011] *Recent Trust Models In Grid*. Journal of Theoretical and Applied Information Technology, Vol. 26 No. 1, pp. 64-68.

[Ermisch J. and Gambetta D., 2006] *Peoples trust: The design of a survey-based experiment*. ISER Working Paper Series, Discussion Paper No. 2216.

[Sarmenta L. F. G., 2002] *Sabotage-tolerance mechanisms for volunteer computing systems*. Future Generation Computer Systems, Vol. 18, No. 4, pp. 561-572.

[Chen C. et al., 2009] *An Approach for Resource Selection and Allocation in Grid Based on Trust Management System*, First International Conference on Future Information Networks, pp. 232236.

[Foster I. et al., 2004] *Brain meets brawn: Why grid and agents need each other*. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Vol. 1, pp. 8-15.

[Kuranowski W. et al., 2008a] *Forming and managing agent teams acting as resource brokers in the grid-preliminary considerations*. International Journal of Computational Intelligence Research, Vol. 4, No. 1, pp. 9-16.

[Kuranowski W. et al., 2008b] *Super- vising agent team an agent-based grid resource brokering system-initial solution*. Proceedings of the Conference on Complex, Intelligent and Software Intensive Systems, pp. 321-326.

[Dominiak M. et al., 2006] *Efficient Matchmaking in an Agent-based Grid Resource Brokering System*. Proceedings of the International Multiconference on Computer Science and Information Technology pp. 327335.

[Resnick P. et al., 2000] *Reputation Systems*. Communications of the ACM, Vol. 43, No. 12, pp. 4548.

[Hoffman K. et al., 2009] *A survey of attack and defense techniques for reputation systems*. ACM Computing Surveys, Vol. 42, No. 1, pp. 1-1.

[Luke Teacy W. T. et al., 2006] *Travos: Trust and reputation in the context of inaccurate information sources*. Autonomous Agents and Multi-Agent Systems, Vol. 12, No. 2, pp. 183198.

[BOINC website] http://boinc.berkeley.edu

[Cuenca-Acuna F. M. et al., 2003] *Autonomous Replication for High Availability in Unstructured P2P systems*. The 22nd International Symposium on Reliable Distributed Systems, pp. 99-108.