A QoS-Aware, Trust-Based Aggregation Model for Grid Federations

Antonello Comi[1], Lidia Fotia[1], Fabrizio Messina[2], Domenico Rosaci[1], and Giuseppe M.L. Sarnè[3]

[1] DIIES, University "Mediterranea" of Reggio Calabria,
Via Graziella, Feo di Vito 89122 Reggio Calabria (RC), Italy
{antonello.comi,lidia.fotia,domenico.rosaci}@unirc.it

[2] Dipartimento di Matematica e Informatica, University of Catania,
V.le Andrea Doria 6, 95125 Catania, Italy
messina@dmi.unict.it

[3] DICEAM, University "Mediterranea" of Reggio Calabria,
Via Graziella, Feo di Vito 89122 Reggio Calabria (RC), Italy
sarne@unirc.it

Abstract. In this work we deal with the issue of optimizing the global Quality of Service (QoS) of a Grid Federation by means of an aggregation model specifically designed for intelligent agents assisting Grid nodes. The proposed model relies on an algorithm, called FGF (Friendship and Group Formation), by which the nodes select their partners with the aim of maximizing the QoS they perceive when a computational task requires the collaboration of several Grid nodes. In the proposed solution, in order to assist the selection of the partners, a suitable trust model has been designed. Since jobs sent to Grid Federations hold complex requirements involving well defined resource sets, trust values are calculated for specific sets of resources. We also provide a theoretical foundation and some experiments to prove that, by means of the adoption of the FGF algorithm suitably supported by the proposed trust model, the Grid Capital (which reflect the global QoS) of the Grid Federation is eventually improved.

1 Introduction

Grid Computing [16] has recently evolved to the Federated Grids architecture [25,38], in which Grid brokers [1,16] and Grid institutions can share resources among different types of Grid infrastructures, resulting as a more flexible approach than that derived by the classical and well known concept of Virtual Organization (VO).

The development of Federated Grids has been strongly encouraged by the increasing complexity of Grid tasks [5] submitted by companies and institutions [45] and supported by the quick and recent advancements of Virtualization technologies [4,26,41]. In particular, such improvements have introduced more flexibility in managing resources, software deployment activities, configuration of

R. Meersman et al. (Eds.): OTM 2014, LNCS 8841, pp. 277–294, 2014.

software platforms, and so on. As a consequence, Grid infrastructures are characterized by a high dynamic, such that companies and institutions are allowed to join or leave different Grid infrastructures in a very easy way, reconfigure their own nodes and resources, modify their own role into the VO, their decisional processes and so on.

In the above scenario, Grid users are encouraged to send job requests characterized by more and more complex requirements, e.g. jobs might require the execution of "inter-grid" workflows, composite services, etc. Consequently, on this base, federated Grid brokers have to deal with complicated tasks of resource allocation by evaluating requirements involving a huge set of *federated resources* in order to satisfy job requirements [8]. This is mainly due to the fact that Grid Federations are intended to exploit the potential *collaboration* between Grid VOs which are able to provide highly specialized, and not trivial, contributions to the result of the expected computation.

Furthermore, federated resources also need to be allocated by paying particular attention to use them in an efficient fashion. For instance, complex job requirements must be satisfied first, and at the same time, choices that might cause unbalanced resources allocations or poor performances should be avoided as much as this is possible. It is even conceivable that the establishment of Grid Federations brings the participants to join with a highly "competitive" scenario. Although competition usually leads better performances, it enhances the presence of possible malicious behaviors. Trust-based systems can help to solve and/or mitigate this problem and, therefore, it is important that competitive Grid Federations are supported by a well suited trust model [7,10,23] in order to quantify the level of performance and reciprocal trust of Grid nodes.

Basing on the considerations above, in this work we propose an approach aiming to maximize the measured "performance" or, in other words, the global Quality of Service (QoS) perceived within the Grid Federation. In this approach we focus on the concept of *resource sets*, i.e. the sets of computational resources characterizing complex jobs in Grid Federations. The proposed solution is based on the use of software agents [17], which, in our approach, manage every node of the Grid Federation. Moreover, the concept of *agent aggregation* (i.e. groups and friendship) is exploited as the basis of collaboration between federated nodes. By combining the use of a Trust model, which provides to compute some measures of reliability and reputation, in turn combined in a unique synthetic trust measure, we propose an algorithm for agent Friendship and Group Formation (FGF) in order to maximize the "global utility", i.e. the *Grid Capital* of the whole Grid federation. Finally, we prove that the execution of the FGF algorithm, which is supported by the dissemination of trust information, allows Grid federated Brokers/Nodes to improve both individual and global satisfaction.

The plan of the paper is as follows. In Section 2 we describe a reference grid scenario for competitive agents. Then, Section 3 introduces our trust model. Section 4 presents the FGF algorithm for forming friendships and groups, while in Section 5 we provide some theoretical results which prove the validity of the proposed FGF algorithm. In Section 6, some experiments devoted to verify the effectiveness of the FGF algorithm are presented. Finally, in Section 7 we discuss some related work and in Section 8 draw our conclusions and introduce our ongoing researches.

2 The Competitive Grid Scenario

This work is based on a generic scenario in which a set N of Grid Nodes provide Grid Services within a Grid Federation [44] to a set C of clients (Grid Users).

Grid Services on Competitive Grids. A Grid Service generally involves the execution of workflows [46] and/or the provisioning of composite services [36], therefore a certain amount of different resources is generally required for their execution, therefore, we assume to classify each Grid Service on the basis of such required resources. More in detail, we model the set of heterogeneous resources which can be found on N as a finite number of R incremental sets of resources, where the last set (i.e. the R-th) includes all the resources belonging to N. Moreover, the Grid scenario we take into consideration is "competitive" and we assume that when the generic client $c_j \in C$ benefits for the service s^r (with $1 \leq r \leq R$) provided by n_i, it will have to pay a fixed price p to n_i based on the set r of consumed resources.

Agents, skills and feedbacks. We also assume that each Grid node is assisted by a software agent [17], supporting it in providing services to Grid Users. Let A be the set of these agents, where the agent $a_i \in A$ is associated with the node $n_i \in N$. Furthermore, each agent a_i is characterized by a "skill" mapping, $\sigma_i(r) \in [0, 1] \subseteq \mathbb{R}$, where 1/0 means the maximum/minimum quality in providing a service requiring that specific set of resources (i.e. r). When the client c_j benefits from the service s^r provided by the agent a_i (i.e. node n_i), it returns a feedback f to a_i, where f is a real value belonging to [0..1], such that 1 means the maximum satisfaction perceived by c_j for s^r and, vice versa, 0 means the minimum perceived satisfaction.

Agent aggregation. In the presented model we rely on the concepts of agent friendships and groups, by supposing that each agent a_i maintains a set F_i of *friend agents*, such that $F_i \subseteq A$, and a set of *groups* $G_i = \{g_{i_1}, \ldots, g_{i_k}\}$ where $\bigcup_{1<l<k} g_{i_l} \subseteq A$.

Agent collaboration for service provisioning. To satisfy the request for a service s^r, the agent a_i can require the *support* of another node n_j (i.e. a_j) that in turn can accept or refuse it[1]. If a_j accepts and it is a friend of a_i or it is in the same group with a_i, this contribution will be provided for free; otherwise, a_j will require the payment of a fee p_s to the agent a_i after the service has been consumed. In order to select the best agents which can collaborate for the service s^r, the agent a_i can ask the *opinion* of other agents. In particular, a_i

[1] We assume that each agent can perform at most X support requests.

can send a request to an agent a_k to obtain a recommendation $rec_j(r)$ about the skill $\sigma_j(r)$ of a_j for a given service s^r. In turn a_k can accept or refuse the request for $rec_j(r)$[2]. If a_k accepts and it is a friend of a_i or it is in the same group with a_i, this recommendation will be provided for free; otherwise, after the recommendation has been provided by a_k, a price p_r should be paid by a_i to a_k. The final choice is performed by the agent a_i based on the Trust model described into Section 3, which takes into account also the *reliability* of the node a_j. It is updated by taking into account the feedback provided by the Grid User who consumed the service.

Looking for agents and groups. In such a scenario, the names of agents, groups and agents belonging to each group have to be appropriately registered, such that their names and locations can be easily retrieved. For this aim we assume that those entities are registered by means of a service named Directory Facilitator (DF), which in turn is published by the various Grid VO (Virtual Organizations) by means of the service infrastructure GIS (Grid Information Services) [11].

3 The Trust Model

In the described grid context, the presence of competitive agents compels us to also consider their possible misbehaviors. For instance, an agent could receive from another agent (i) less resources from those corresponding to its actual skill coefficient σ or (ii) an unfairly recommendation about the skills of a third agent. To manage such misbehaviors, in multi-agent environments a common solution is represented by trust systems [24]. To this purpose, in the proposed model, for each agent a_j to which it interacted in the past, each agent a_i maintains a triple of values (α, β, γ), respectively called *Reliability*, *Honesty* and *Reputation* that belong to $[0, 1] \in \mathbb{R}$.

Reliability $(\alpha_{ij}(r))$. The first value, denoted by $\alpha_{ij}(r)$, is the *reliability* of a_j in providing a set r of resources, and represents how much a_i trusts a_j in its capability to provide resources for a service s^r. More in detail, for each feedback f received by a_i for a service s^r, if a_i was supported in its task by a_j, the feedback f includes a contribution f_j^* due to a_j, that will be assigned to a_j. Obviously, if a_i completely delegated a_j to provide s^r, then it will be $f_j = f = f_j^*$. Finally, the reliability is computed as the arithmetical mean of all the feedback received by a_i for all the services which required the set of resources r and the collaboration of a_j:

$$\alpha_{i,j}(r) = \begin{cases} \frac{1}{l}\sum_{m=1}^{l} f_j^m & \text{\textit{if} } l \neq 0 \\ \\ \overline{\alpha} & \text{\textit{if} } l = 0 \end{cases} \tag{1}$$

[2] We assume that each agent can perform at most Y recommendation requests.

Honesty ($\beta_{i,j}(r)$). The second value, denoted by $\beta_{i,j}(r)$, is referred as the honesty of a_j in giving a recommendation to a_i about the capability of another agent to provide a set r of resources. It is computed by comparing the feedbacks $f_{x_1},...,f_{x_l}$ obtained by a_i for some agents a_{x_k} suggested by a_j. In other words, it is the arithmetical mean of all the difference between each f_k and the associated recommendation $rec_j^{\,l}$ provided by a_j about some agents a_{x_l}. More formally:

$$\beta_{i,j}(r) = \begin{cases} \frac{1}{l}\sum_{m=1}^{l} |rec_j^m(r) - f_{x_m}(r)| & \mathit{if}\ l \neq 0 \\ \overline{\beta} & \mathit{if}\ l = 0 \end{cases} \quad (2)$$

Reputation ($\gamma_{ij}(r)$). The last value is the *reputation* of a_j with respect to the set of resources r, denoted by $\gamma_j(r)$, representing how much the agents interrogated by a_i estimated the capability of a_j about the resource set r provided by a_j. The reputation $\gamma_{ij}(r)$ is computed as the mean of all the recommendations that each other agent provided to a_i about a_j on r weighted by its *honesty* value. More formally:

$$\gamma_{i,j}(r) = \begin{cases} \frac{1}{l}\sum_{m=1}^{l} rec_j^m(r)\beta_{im}(r) & \mathit{if}\ l \neq 0 \\ \overline{\gamma} & \mathit{if}\ l = 0 \end{cases} \quad (3)$$

Note that in the equations described above, the values $\overline{\alpha}$, $\overline{\beta}$ and $\overline{\gamma}$, called "cold start" values, are used in the case any interaction with other agents occurred in the past for a_i with respect to the considered set of resources r as, for instance, for the new coming agents.

Finally, it is possible to compute the synthetic measure of trust of a_i about an agent a_j with respect to the set of resources r, denoted by $\tau_{i,j}(r)$, as:

$$\tau_{i,j}(r) = \delta_i \cdot \alpha_{i,j}(r) + (1 - \delta_i) \cdot \gamma_j(r)$$

where $\delta \in [0, 1] \subset \mathbb{R}$ is used to weight the relevance assigned by a_i to the reliability with respect to the reputation.

4 The Friendship and Group Formation (FGF) Algorithm

As described in Section 2, when an agent a_i asks for a contribution or a recommendation to another agent a_j which is a friend or a member of its same groups, it will be for free. The difference is that in the first case a_i and a_j mutually accepted to become friends in the past, while in the latter a_j could be only a group mate and not also a friend for a_i.

In the above context, a_i can build two sets of preferred agents for each set of resources r, namely:

- a set PC_i^r storing the *preferred contributors* agents which a_i interacted in the past for a contribution falling in r and having (i) the X highest trust values $\tau(r)$ and (ii) a trust value greater than the threshold τ^{min}.

– a set PR_i^r storing the *preferred recommenders* agents which a_i interacted in the past for a suggestion falling in r and having (i) the Y highest honesty values $\beta(r)$ and (ii) a honesty value greater than the threshold β^{min}.

Basing on the definition of trust and honesty provided in Section 3, in order to maximize the performance of the services provided by a_i in collaboration with other agents, its own sets F_i (friends) and $g \in G_i$ (groups) should only include the agents belonging to PC_i^r and PR_i^r for all the set of resources:

$$\bigcup_{r\in R} (PC_i^r \cup PR_i^r) = F_i \bigcup (\bigcup_{g\in G_i} g) \tag{4}$$

In order to adopt a convenient notation, we will use the following definitions:

$$PA_i^r = PC_i^r \cup PR_i^r; \qquad AG_i = \bigcup_{g\in G_i} g$$

and therefore Equation 4 can be written as

$$\bigcup_{r\in R} PA_i^r \ = \ F_i \bigcup AG_i \tag{5}$$

In particular, when Equation 5 is not verified, we have to take into account the two cases described below.

Loss of performance (L): $\left(\bigcup_{r\in R} PA_i^r\right) - \left(F_i \bigcup AG_i\right) \neq \ \emptyset$, i.e. there are some agents belonging to the set $\bigcup_{r\in R} PA_i^r$ but not to the set $F_i \bigcup AG_i$. The consequence is represented by the possible *loss of performance*, in providing Grid Services, due to the selection of one of these agents. The issue above can be measured by calculating, for the set $\bigcup_{r\in R} PC_i^r$ (resp. $\bigcup_{r\in R} PR_i^r$), the difference $|\tau_{i,j}(r^*) - \tau_{i,alt_j}(r^*)|$, where r^* is the resource sets in which a_j is a preferred contributors (resp. preferred recommender) agent and alt_j is the agent in $F_i \bigcup AG_i$ having the best trust (resp. honesty) value on r^*. Therefore, we compute the factor *Loss of Performance* (L_i), for an agent i, as the average of the sum of the two contributions described above:

$$L_i = \frac{L_i^{(\tau)} + L_i^{(\beta)}}{2}$$

where

$$L_i^{(\tau)} = \frac{\sum_{j\in\left(\bigcup_{r\in R} PC_i^r - F_i \bigcup AG_i\right)} \left(\tau_{i,j}(r^*) - \tau_{i,alt_j}(r^*)\right)}{\| \bigcup_{r\in R} PC_i^r - F_i \bigcup AG_i\|}$$

and

$$L_i^{(\beta)} = \frac{\sum_{j\in\left(\bigcup_{r\in R} PR_i^r - F_i \bigcup AG_i\right)} \left(\beta_{i,j}(r^*) - \beta_{i,alt_j}(r^*)\right)}{\| \bigcup_{r\in R} PR_i^r - F_i \bigcup AG_i\|}$$

If a_j is a preferred contributor (recommender) agent on more resources sets, r^* will be the set having the highest trust (honesty) value, then the factor $L_i^{(\tau)}$ (resp. $L_i^{(\beta)}$) is obtained by computing the average of all these contributions.

Additional Cost (C): $\left(F_i \bigcup AG_i\right) - \left(\bigcup_{r\in R} PA_i^r\right) \neq \emptyset$. It means that some agents belong to the set $F_i \bigcup AG_i$ but not to the set $\bigcup_{r\in R} PA_i^r$. In this case we measure the ratio of agents that will be never contacted by a_i to obtain help for free by computing the factor *Additional Cost* as:

$$C_i = \frac{\parallel F_i \bigcup AG_i - \bigcup_{r\in R} PA_i^r \parallel}{\parallel F_i \bigcup AG_i \parallel}$$

As a consequence, we can measure the "disadvantage" of a_i as the average of the sum of the factors L_i^τ, L_i^β and C_i as follows:

$$D_i = \frac{L_i^\tau + L_i^\beta + C_i}{3} \tag{6}$$

4.1 The FGF Algorithm

Here we provide an algorithm to be executed by each agent a_i in order to minimize, during various *epochs*, the disadvantage D_i provided by Equation 6.

We assume that the period of time between two consecutive epochs is set to a pre-fixed value T and that, in each epoch, some preferred agents join with the set $F_i \bigcup AG_i$ to replace agents having the worst trust or honesty values. The FGF algorithm is composed by two parts: the former (*Task A*) is a procedure designed to improve the coefficient D_i given by the Equation 6 by means of some requests which shall be initiated by the agent executing the procedure itself. The second (*Task B*) is designed to manage the requests coming from the other agents due to execution of the *Task A*.

Task A

Each agent a_i periodically executes the task A to obtain the friendship or the membership in a group of G_i for those agents belonging to the set $\bigcup_{r\in R} PA_i^r$ but not yet to the set $F_i \bigcup AG_i$. The Task A is composed by the following ordered sequence of steps (refer to Figure 1):

1. The sets $F_i \bigcup AG_i$, and $\bigcup_{r\in R} PA_i^r$ are computed (Figure 1, step 1).
2. A friendship request is sent to each agent $a_j \in \left(\bigcup_{r\in R} PA_i^r - F_i \bigcup AG_i\right)$ (Figure 1, step 2).
3. If a_j accepts the friendship request, then it is added to F_i (Figure 1, step 3).
4. If a_j does not accept the friendship request, then a_i executes the following steps:
   (a) the set G_j of all the groups having a_j as a member is required by a_i to the DF (for a description of the DF see Section 2).
   (b) for each group $g \in G_j$, a_j computes the disadvantage D_i^*.
   (c) a joining request is sent to the group $g \in G_j$ (Figure 1, step 4) such that $D_i^* < D_i$ and D_i^* is minimum.

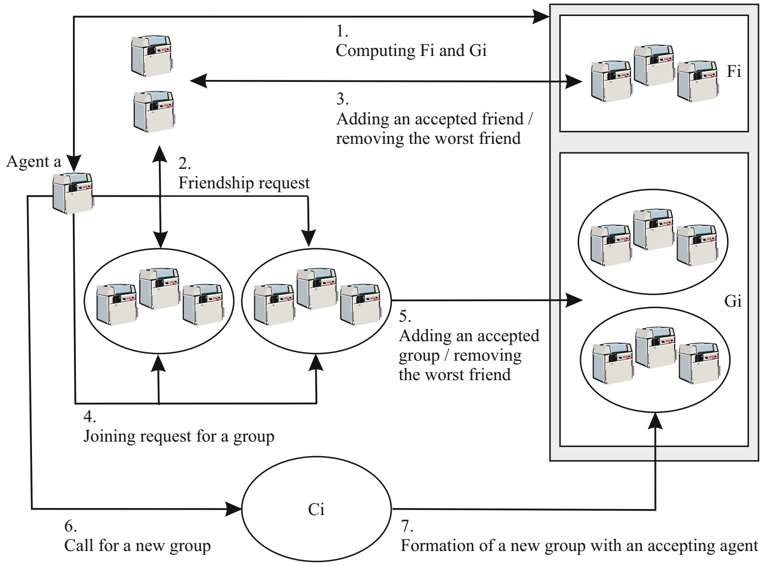

Fig. 1. The task A of an agent

(d) If g accepts the membership request then g is added to G_i (Figure 1, step 5), otherwise a_j is added to a set C_i.

5. If C_i is not empty, then a_i sends a *call for a new group* (Figure 1, step 6) to all the agents belonging to it. If some agents agree with constituting a new group, then it is formed (Figure 1, step 7) and registered to the DF.
6. Whenever an agent a_j is added to the set F_i, then an agent a_k, the worst friend, is removed from F_i (Figure 1, steps 3 and 5). The agent a_k is selected as follows:
   – if $a_j \in PC_i^r$, then select $a_k \notin \left(\bigcup_{r \in R} PA_i^r\right)$ having the worst trust value $\tau_{i,k}(r)$ or
   – if $a_j \in PR_i^r$, then select $a_k \notin \left(\bigcup_{r \in R} PA_i^r\right)$ having the worst honesty value $\beta_{i,k}(r)$.

Task B

Task B is a set of three subtasks designed to manage the friendship requests of the other agents, as well as the requests of joining that other agents send to groups with which a_i is joined or is a leader (administrator). The execution of one of these subtasks depends both on the role of the agent and on the nature of the received request, as specified in the following.

– *Friendship Request.* When such a request coming from an agent a_j arrives to an agent a_i then a_i:
   1. computes a new disadvantage D_i^* by adding the agent a_j to the set F_i and removing an agent a_k as described by step 5 of Task A (Figure 2, Step 1).

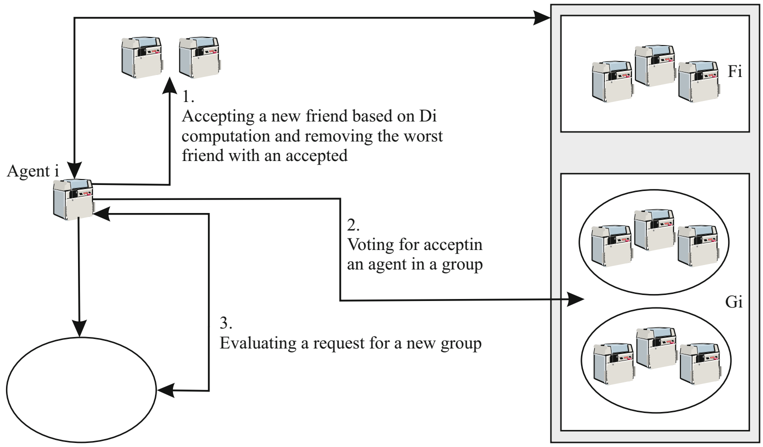

Fig. 2. The task B of an agent

2. will accept the request of a_j it $D_i^* \leq D_i$. Otherwise, the request will be refused.

– *Membership Request.* When such a request coming from an agent a_j arrives to the administrator of a group g (Figure 2, Step 2), it requires a vote (positive or negative) to all the agents belonging to g. The request will be accepted by majority, otherwise it will be refused. Each agent a_k will give a positive vote if the insertion of a_j in the group g will not increase the disadvantage D_k (Figure 2, step 3).
– *Call for a new group.* Such a request coming from an agent a_j is accepted by a_i if the insertion of a_j in the set $F_i \bigcup AG_i$ does not increase the disadvantage D_i (Figure 2, step 3).

5 Theoretical Results

In this Section we theoretical prove the benefits coming from the adoption of the described friendships and groups model as well as the FGF algorithm presented in Sections 3 and 4.

To this purpose, let the *Grid Capital* (GC) be the mean value of all the contributions $(1 - D_i)$ given by each agent a_i, computed as $GC = \frac{\sum_{a_i \in A}(1-D_i)}{\|A\|}$. The *Grid Capital* increases at each iteration of the FGF algorithm because the FGF is able to optimize the global grid utility by searching new relationships among agents.

Theorem 1. *The Grid Capital increases at each iteration of the FGF algorithm.*

Proof. Taking into account the activities performed by the agent a_i at each iteration of the Task A (Section 4), we obtain that D_i: (i) increases if one or more preferred contributor or recommender agents accept its joining request; (ii)

is unvaried; (iii) decreases if one or more preferred contributor or recommender agents exit from F_i or from a group belonging to G_i. Therefore, let a_j be a preferred contributor or recommender agent in a resources set r^*. The agent a_j will exit from F_i or from a group belonging to G_i only if a_i is not one of its preferred contributors or recommenders agents and it implies that D_j decrements by 1, while D_i will increase of $\tau_{i,j}(r^*) - \tau_{i,alt_j}(r^*)$, that is lesser than 1, due to the replacement of a_j with the best alternative alt_j . Consequently, we can derive that the sum of all the agent disadvantages decreases at each iteration, while the sum of all the contributions $(1 - D_i)$ increases and this proves the Theorem 1.

In order to specify the global trustworthiness that an agent a_i receives from the whole agent community in a resources set r, we assume that: (i) let the *Merit* of an agent a_i in the resources set r (φ_i^r) be the number of agents for which a_i is preferred as contributor or recommender; (ii) let the *Expected Gain* of an agent a_i (ω_i) be the expected gain of a_i at a given step; (iii) let $P_i(x)$ be the expected value of the probability distribution such that $\omega_i = x$.

Based on this assumptions, we can state that if a_i and a_j are two agents, such that $\varphi_i^r < \varphi_j^r$ at a given iteration, then the number nc_i of clients contacting a_i for a service request falling in the resources set r will be lesser than the number nc_j of clients contacting a_j. It seems to be a reasonable consequence because if $\varphi_i^r < \varphi_j^r$, this implies that the global satisfaction of the agent community for the a_i performances is lesser than for a_j. Since the global satisfaction of the agent community is based on the clients feedbacks, it is reasonable to suppose that similarly also the clients will prefer to contact a_i instead of a_j. In other words, the choices of the clients reflect as a mirror the choices of the agents. It is true when the trustworthiness of an agent, represented by the number of other agents that consider it as a preferred interlocutor, actually capture its expertise. The mirror assumption can be considered as much valid as (i) the agent trust models are strictly based on the clients' feedbacks, similarly to that presented in Section 3 and (ii) reflects the real situation as much as the adopted trust model is able to capture the actual expertises of the agents.

Theorem 2. *For each pair of agents a_i and a_j, such that at each iteration $\varphi_i^r < \varphi_j^r$, the expected gain ω_i will be lesser than the expected gain ω_j.*

Proof. Supposing as valid the theorem statement, it implies that the number nc_i of clients contacting a_i for a service request related to r will be lesser than the number nc_j of clients contacting a_j. Consequently, the probability P_i that a contribution or a recommendation related to r can be requested by other x_i agents to a_i is lesser than the corresponding probability P_j that x_j agents can require it to a_j. Similarly, the expected number y_i of agents contacted by a_i for the resources set r is greater than the expected number y_j of agents contacted by a_j; it is due to the high probability that the expertise of a_i is smaller than that of a_j. For sake of simplicity, suppose that a contribution or a recommendation have the same price p^*, thus at the end of the current iteration the expected gain ω_i is $u_i \cdot p + x_i \cdot p^* - y_i \cdot p^*$, that is smaller than the corresponding gain ω_j for the agent a_j and this proves the Theorem 2.

6 Experiments

In this Section we present the results of some experiments by which we study the behavior of the proposed model as a function of the parameters previously introduced. More in detail, the results shown in this Section have been obtained by a set of simulations performed on the Octave numerical tool [22] and based on a set of parameters that we summarize in Table 1.

Table 1. Simulation parameters

Simulation Parameter	**Value**
N_{nodes}	1000
N_{grids}	10
r sets	$\{r_1, r_2, r_3\}$
No. of HP Grids	G_1–G_4
No. of MP Grids	G_5–G_7
No. of LP Grids	G_8–G_{10}
H.P. feedbacks, (range of gen. values)	$\{0.6 \ldots 0.8\}$
M.P. feedbacks, (range of gen. values)	$\{0.4 \ldots 0.6\}$
L.P. feedbacks, (range of gen. values)	$\{0.1 \ldots 0.4\}$
Recommendations (range of gen. values)	$\tau \pm (0.1\tau)$
Average no. of generated Feedbacks (per step)	$\{\frac{N}{10} \ldots \frac{N}{2}\}$
Average no. of generated Reccomendations (per step)	$\{\frac{N}{10} \ldots \frac{N}{2}\}$
$(\overline{\alpha}, \overline{\beta}, \overline{\gamma})$	$(0.5, 0.5, 0.5)$
Average no. of friends (random)	$\simeq \sqrt{N}$
Average no. of groups	$\simeq \frac{N}{2}$
Average level of group membership (random)	$\simeq \frac{\sqrt{N}}{2}$

```
for T = 1 : END
    advance_epoch() // It serves as key for collected data
    simulate_feedbacks() // Parallel
    simulate_recc() // Parallel
    update_tau_for_all_nodes() // Parallel
    update_disadvantage_for_all_nodes() // Parallel
    for a = 1 : N
      task_A(a); // Functions of Task B are consequently invoked.
    endfor
endfor
```

Fig. 3. Simulation flow

We simulated (1) a Grid federation composed by 1000 nodes equally distributed into 10 different Grid Virtual Organizations, G_1–G_{10}, each one exposing different performance levels, i.e. High performance (HP), Middle Performance

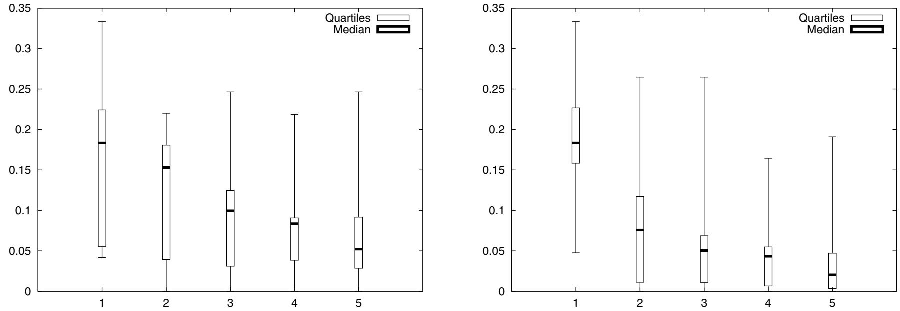

Fig. 4. Disadvantage (D). Min, 1st Quartile, Median, 3th quartile, Max. Left: X=Y=10 Right: X=Y=40, $\tau^{min} = \beta^{min} = 0.2$

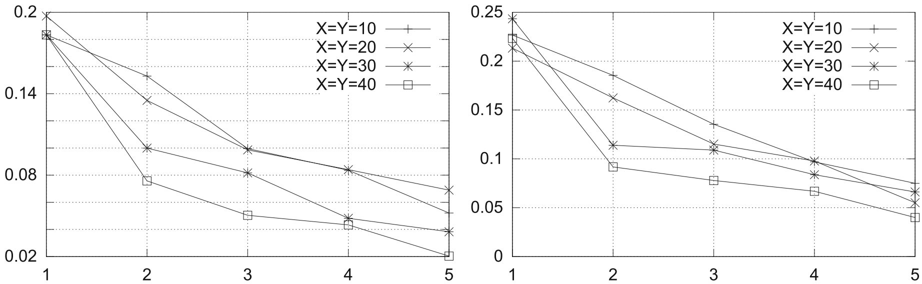

Fig. 5. Median value of disadvantage, Left: $\tau^{min} = \beta^{min} = 0.2$. Right: $\tau^{min} = \beta^{min} = 0.5$

(MP), and Low Performance (LP). The term “performance” is used in this context to indicate the level of simulated reliability of the considered Grid infrastructure or, in other words, of the related nodes providing services. Therefore, the different levels of performance are simulated by generating different values for feedbacks – and, on the consequence, different values of recommendations – for different services.

Since we were interested to confirm the theoretical results provided into the previous Section about a lowering of the Disadvantage (D) (i.e. a rices of the average *Grid Capital* $GC = \frac{\sum_{a_i \in A}(1-D_i)}{\|A\|}$) caused by the execution of the FGF algorithm presented into Section 4, we did not make any distinction between resource sets among different Grids. Then we assumed, as shown into Table 1, that all the Grids have the same “bag” of resource sets, r_1, r_2, r_3. Moreover, Table 1 also shows the average number of friends and groups. In particular, the “average level of group membership” indicates the average number of groups to which the Grid nodes have been attached during the network generation.

To let the reader better understand how we simulated the proposed model, we provide in Figure 3 a very simple listing of pseudo code, which represents the basic simulation flow we employed for the experiments. The code is serial for that concerning the execution of *taskA*, but we have made parallel the code for which no critical races can occur, indicating them as *Parallel* into Figure 3 [3].

We report into Figure 4 the median values, quartiles and outliers of the Disadvantage for a set of simulations for which we set $\tau^{min} = \beta^{min} = 0.2$, i.e. the minimum value of trust and honesty to select the set PC and PR (see Section 3). We report only the first five steps of the simulation because the trend stabilizes very quickly and, although the initial groups and friends are initially random, after the first step of execution of the Task A the average disadvantage becomes very low. While on the left part of Figure 4 we present results for $(X = Y = 10)$, where X and Y are the maximum size of the sets PC and PR (see Section 3), the results shown on the right concern $(X = Y = 40)$. We observed that the median value of the disadvantage has a downward trend, thus confirming the theoretical results presented into the previous Section 5. More important, we remark that, as the sets PC and PR grow in size (from $(X = Y = 10)$ – left part of Figure 4 to $(X = Y = 40)$ – right part of Figure 4), the median (so the quartiles) assumes lower values very quickly, which is an expected behavior.

Furthermore, the behavior explained above is better described by results provided into Figure 5, on which we plotted only the median value of the Disadvantage, and X and Y ranging from 10 to 40 by steps of 10. Moreover, by comparing the results on the left part of Figure 5 with those shown in the right part of Figure 5, we observed that the more selective is the parameter τ^{min} (*i.e.* β^{min}) (i.e. the minimum value of trust (i.e. honesty) to put a node into the set PC (i.e. PR), the greater will be, in average, the disadvantage. Also this behavior seems to be conform with the described model and theoretical results.

Summarizing, the results of the experiments presented in this Section clearly show as the proposed model is correct and the execution of the FGF algorithm, supported by the adopted trust model, it is effectively capable to allow Grid federations to improve their provided/perceived QoS.

7 Related Work

The various aspects related to the partner/node selection and collaboration issues in the context of self-interested agents and grid systems have been dealt in a large number of models and architectures. Their overall discussion would require too much space and, therefore, the examined approaches will be only those that, to the best of our knowledge, come closest to our proposal.

In the literature different metrics have been proposed to select the most appropriate partners, for instance by exploiting local decision and models [30,37] or by promoting agent interactions to realize a distributed social control mechanism for evaluating other agents or their provided services [14]. Many of such models consider direct observations and/or communications with other agents and different criteria as trust, reputation, provided QoS, etc. In this context,

[3] We used the package *parallel* provided with Gnu Octave.

the concept of *belief* can be considered as a situational awareness, and its modification can require to select the most appropriate providers for information. To this purpose, in multi-agent systems the beliefs of the agents might consider the preferred agents to obtain suitable information [2,18], where the preferred agents could be selected, for instance, on the basis of trustworthiness [2] or statistical [18] criteria. Note that these techniques usually result computational expansive [28,35] and therefore some approximate and less expensive solutions have been proposed in the context of specific beliefs [9,27].

To solve coalition formation issues among self-interested agents, negotiation mechanisms (requiring peer-to-peer communications) can be used to find the best candidates to join with. The Contract Net Protocol (CNP) [12,43] is a fully automated negotiation protocol where each agent can be an initiator or a participant of a call for proposal and where only the best participants bids on that call are selected by the initiator. This (relatively) simple partner selection mechanism might result computationally expensive on large-scale systems due to the message approach and, moreover, it does not guarantee the real execution of the "contract". However, the CNP has been embedded into the Transportation Cooperation Net (TRACONET) [42] for a vehicle routing application according to the Foundation for Intelligent Physical Agents (FIPA) standard. Another negotiation partner selection scheme is the Adaptive Decision Making Framework (ADMF) [3]; it has been designed for systems where agents, assumed to be cooperative, share global goals to be maximized and allows a dynamic adjustment of agents relationships, although also this proposal has a low scalability in presence of large-scale systems.

In a multi-agent system the coalition formation provides to partition agents in groups in order to optimize groups and/or agents utility. The agents partition activity, might be modeled as a function game [42] involving *a*) the generation of the coalition structure, *b*) the solution of optimization problems (for each coalition) and *c*) the pay-off distribution. Activities *a* and *b* provide to search suitable partnerships from a set of cluster of agents, while the last activity distributes the coalition gain among the participants and, as important *collateral* effects, it promotes the agents' collaboration and the stability of the coalition. Recently, some proposals adopted trust in competitive agent systems [21,39], for instance, to constitute clusters of agents [6,20] and for generating recommendations in social network contexts [13] or to detect group of actors in a competitive social community [31,32,40].

Summarizing, none of the cited proposals deal with the issue to improve the social capital of the agent community on the basis of a meritocracy criterion. On the contrary, such approaches exploit trust measures to provide an agent with suggestions about the best agents to contact as fruitful interlocutors, but without to face the issue of the possible advantages for its community. Differently, our proposal introduce a meritocratic principle in order to obtain such an advantage, also by encouraging the actors to assume correct behaviors to improve their reputation.

On the other hand, employing software agents into Grid systems has always been a subject of research [15]. Several works in the literature focused on the optimization of various aspects of Grid jobs scheduling and resources allocation. Most of them rely on the adoption of economical models as, for instance, in [7]. In [29] a strategy for optimizing the QoS into the Grid is presented. The work is based on a distributed iterative algorithm behind a mathematical model. The authors mainly deal with task optimization and resource optimization, by means of software agents. Even their goal is maximizing the global utility of the system, their approach is not distributed, indeed the different agents collaborate to perform optimization activities for the whole system. Authors of [10] analyze the interactions between Grid user agents and the Grid providers in order to maximize the whole utility of all Grid users. They propose a price-based resource allocation model by defining a *Grid User Utility* as a function of the user's allocated resources by using a nonlinear optimization theory, in order to incorporate Grid resource capacity constraint and job completion times. Nevertheless, they do not deal with the the heterogeneity of resources, and do not rely on the concept of meritocracy (reliability and trust) to improve the overall QoS.

Authors of [19] combined the principles and the concepts found in social networks to design decentralized and adaptive resource discovery approach in complex Grid systems. Experimental results show as the relationship among clusters can improve the resource discovery processes, allows different resource distributions and user request patterns to a better adaptation, but the approach lacks of a component permitting to improve the social capital of the agent (node) community by improving meritocracy.

8 Conclusions and Future Work

In this paper, we presented an agent based model to optimise the global QoS of a "competitive" Grid Federation, on which computational nodes are supported by intelligent agents, which manage friendships and group memberships.Our proposal focused on the concepts of (*i*) computational *resource sets* characterising jobs in Grid Federations, (*ii*) *agent aggregation* (i.e. friendships and group memberships) as basis of collaboration between federated nodes, which, in turn, are supported by (*iii*) a *trust* model conceived to compute a unique synthetic trust measure from reliability, honesty and reputation measures. We designed an algorithm, called Friendship and Group Formation (FGF), which allows Grid nodes to select their partners (friends and group memberships) in order to improve the global QoS. For this aim the algorithm uses the trust information to compute two measures, the (*i*) *disadvantage* (D), which represents a local indication of the QoS that the single node is able to provide to the other Grid nodes, and the (*ii*) *Grid Capital* (GC), which is a global index, telling us how well the Brokers/Nodes of the Grid Federations can work together when a computational task requires an inter-Grid collaboration. The validity of the proposed model has been supported by theoretical evidences and by some experimental results, by which we have shown that the adoption of the FGF algorithm, suitably supported by the proposed trust model, the Grid Capital (which reflects the global

QoS) of the Grid Federation is effectively improved. In our ongoing research, we plan to better study the influence of several parameters characterising our model also by considering very large Grid Federations. Moreover, we plan to compare the performance of the FGF algorithm with other similar approaches which are based on aggregations and trust information. For this aim, we will possibly use ComplexSim [33,34], which is a C-based complex network parallel simulator written by some of the authors.

Acknowledgements. This work has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research and by the Program "Programma Operativo Nazionale Ricerca e Competitività" 2007-2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research.

References

1. Bandieramonte, M., Di Stefano, A., Morana, G.: An ACO inspired strategy to improve jobs scheduling in a grid environment. In: Bourgeois, A.G., Zheng, S.Q. (eds.) ICA3PP 2008. LNCS, vol. 5022, pp. 30–41. Springer, Heidelberg (2008)
2. Barber, K.S., Kim, J.: Soft security: Isolating unreliable agents from society. In: Falcone, R., Barber, S.K., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 224–233. Springer, Heidelberg (2003)
3. Barber, K.S., Martin, C.E.: Dynamic reorganization of decision-making groups. In: Proc. of the 5th Int. Conf. on Autonomous Agents, pp. 513–520. ACM (2001)
4. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. ACM SIGOPS Operating Systems Review 37(5), 164–177 (2003)
5. Boghosian, B., Coveney, P., Dong, S., Finn, L., Jha, S., Karniadakis, G., Karonis, N.: Nektar, spice and vortonics: using federated grids for large scale scientific applications. Cluster Computing 10(3), 351–364 (2007)
6. Buccafurri, F., Comi, A., Lax, G., Rosaci, D.: A trust-based approach to clustering agents on the basis of their expertise. In: Proc. of Advances in Agent and Multi-Agent Systems 2014, AMSTA 2014. ACM Press (2014)
7. Buyya, R., Abramson, D., Giddy, J., Stockinger, H.: Economic models for resource management and scheduling in grid computing. Concurrency and computation: practice and experience 14(13-15), 1507–1542 (2002)
8. Buyya, R., Ranjan, R.: Special section: Federated resource management in grid and cloud computing systems. Future Generation Computer Systems 26(8), 1189–1191 (2010)
9. Chopra, S., Parikh, R., Wassermann, R.: Approximate belief revision preliminary report. Journal of IGPL (2000)
10. Chunlin, L., Layuan, L.: Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. Applied Intelligence 25(2), 147–158 (2006)
11. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid information services for distributed resource sharing. In: 2001 Proc. 10th IEEE Int. Symp. on High Performance Distributed Computing, pp. 181–194. IEEE (2001)

12. Davis, R., Smith, R.G.: Negotiation as a metaphor for distributed problem solving. Artificial intelligence 20(1), 63–109 (1983)
13. De Meo, P., De Meo, A., Rosaci, D., Ursino, D.: Recommendation of reliable users, social networks and high-quality resources in a social internetworking system. AI Communications 24(1), 29–50 (2011)
14. Decker, K., Sycara, K., Williamson, M.: Middle-agents for the internet. In: IJCAI (1), pp. 578–583 (1997)
15. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: Why grid and agents need each other. In: Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multiagent Systems, vol. 1, pp. 8–15. IEEE Computer Society (2004)
16. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. International Journal of High Performance Computing Applications 15(3), 200–222 (2001)
17. Franklin, S., Graesse, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Jennings, N.R., Wooldridge, M.J., Müller, J.P. (eds.) ECAI-WS 1996 and ATAL 1996. LNCS, vol. 1193, Springer, Heidelberg (1997)
18. Fullam, K.K., Barber, K.S.: Using policies for information valuation to justify beliefs. In: Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 404–411. IEEE (2004)
19. Gao, L., Ding, Y.S., Ying, H.: An adaptive social network-inspired approach to resource discovery for the complex grid systems. International Journal of General Systems 35(3), 347–360 (2006)
20. Garruzzo, S., Rosaci, D.: Agent clustering based on semantic negotiation. ACM Transactions on Autonomous and Adaptive Systems 3(2) (2008)
21. Garruzzo, S., Rosaci, D., Sarné, G.M.L.: Integrating trust measures in multi-agent systems. International Journal of Intelligent Systems 27(1), 1–15 (2012)
22. GNU Octave, http://www.gnu.org/software/octave/
23. Yuan, L.-L., Zeng, G.-S., Jiang, L.-L., Jiang, C.-J.: Dynamic level scheduling based on trust model in grid computing. Chinese Journal of Computers 7, 23 (2006)
24. Huynh, T.D.: Trust and reputation in open multi-agent systems. PhD thesis, University of Southampton (2006)
25. Katia, L., Eduardo, H., Ignacio, M.L.: A decentralized model for scheduling independent tasks in federated grids. Future Generation Computer Systems 25(8), 840–852 (2009)
26. Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A.: KVM: the linux virtual machine monitor. In: Proc. of the Linux Symp., vol. 1, pp. 225–230 (2007)
27. Koller, D., Lerner, U., Angelov, D.: A general algorithm for approximate inference and its application to hybrid bayes nets. In: Proc. of the 15th conf. on Uncertainty in artificial intell., pp. 324–333. Morgan Kaufmann Pub. (1999)
28. Lerner, U., Segal, E., Koller, D.: Exact inference in networks with discrete children of continuous parents. In: Proc. of the 17th Conf. on Uncertainty in Artificial Intell., pp. 319–328. Morgan Kaufmann Pub. (2001)
29. Li, C., Li, L.: Utility-based qos optimisation strategy for multi-criteria scheduling on the grid. Journal of Parallel and Distributed Computing 67(2), 142–153 (2007)
30. Maximilien, E.M., Singh, M.P.: Agent-based trust model involving multiple qualities. In: Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems, pp. 519–526. ACM (2005)
31. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A distributed agent-based approach for supporting group formation in P2P e-learning. In: Baldoni, M., Baroglio, C., Boella, G., Micalizio, R. (eds.) AI*IA 2013. LNCS, vol. 8249, pp. 312–323. Springer, Heidelberg (2013)

32. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: HySoN: A distributed agent-based protocol for group formation in online social networks. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) MATES 2013. LNCS, vol. 8076, pp. 320–333. Springer, Heidelberg (2013)
33. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: An smp-aware complex network simulation framework. In: 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 861–866. IEEE (2012), doi:10.1109/CISIS.2012.102
34. Messina, F., Pappalardo, G., Santoro, C.: Complexsim: a flexible simulation platform for complex systems. International Journal of Simulation and Process Modelling 8(4), 202–211 (2013), doi:10.1504/IJSPM.2013.059417
35. Nebel, B.: How hard is it to revise a belief base? Springer (1998)
36. Papazoglou, M.P.: Service-oriented computing: Concepts, characteristics and directions. In: Proc. of the 4th Int. Conf. on Web Information Systems Engineering, WISE 2003, pp. 3–12. IEEE (2003)
37. Park, J., Barber, K.S.: Information quality assurance by lazy exploration of information source combinations space in open multi-agent systems. J. UCS 11(1), 193–209 (2005)
38. Ranjan, R., Chan, L., Harwood, A., Karunasekera, S., Buyya, R.: Decentralised resource discovery service for large scale federated grids. In: IEEE International Conference on e-Science and Grid Computing, pp. 379–387. IEEE (2007)
39. Rosaci, D.: Trust measures for competitive agents. Knowledge-based Systems (KBS) 28(46), 38–46 (2012)
40. Rosaci, D., Sarné, G.M.L.: Matching users with groups in social networks. In: Zavoral, F., Jung, J.J., Badica, C. (eds.) IDC 2013. SCI, vol. 511, pp. 45–54. Springer, Heidelberg (2013)
41. Rosenblum, M.: Vmwar's virtual platform. In: Proceedings of hot chips, pp. 185–196 (1999)
42. Sandhlom, T.W., Lesser, V.R.T.: Coalitions among computationally bounded agents. Artificial intelligence 94(1), 99–137 (1997)
43. Smith, R.G., Davis, R.: Frameworks for cooperation in distributed problem solving. IEEE Trans. on Systems, Man and Cybernetics 11(1), 61–70 (1981)
44. Vázquez, T., Huedo, E., Montero, R.S., Llorente, I.M.: Evaluation of a utility computing model based on the federation of grid infrastructures. In: Kermarrec, A.-M., Bougé, L., Priol, T. (eds.) Euro-Par 2007. LNCS, vol. 4641, pp. 372–381. Springer, Heidelberg (2007)
45. Yu, J., Buyya, R.: A novel architecture for realizing grid workflow using tuple spaces. In: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing, pp. 119–128. IEEE (2004)
46. Yu, J., Buyya, R.: A taxonomy of workflow management systems for grid computing. Journal of Grid Computing 3(3-4), 171–200 (2005)