A partnership-based approach to improve QoS on Federated Computing Infrastructures

A partnership-based approach to improve QoS on Federated Computing Infrastructures

Antonello Comi[a], Lidia Fotia[a], Fabrizio Messina[b], Domenico Rosaci[a,*], Giuseppe M. L. Sarné[c]

[a] *DIIES, Università "Mediterranea" of Reggio Calabria, Italy*
[b] *DMI, University of Catania, Italy*
[c] *DICEAM, Università "Mediterranea" of Reggio Calabria, Italy*

[*] DIIES, Università "Mediterranea" of Reggio Calabria, Via Graziella, Loc. Feo di Vito, 89122, Reggio Calabria, Italy, e-mail: domenico.rosaci@unirc.it

Abstract

In this work we present an approach aimed at maximizing the global QoS perceived within a large-scale federation of computing infrastructures. This approach exploits the combination of (*i*) a trust model for a network of software agents, designed to assist federated computing nodes, and (*ii*) a decentralized procedure which leads to the formation of coalitions between them. The proposed solution is based on a generic SLA-based federated architecture and the concept of "Global Capital" which reflects the global QoS offered by the federation. Finally, a number of experimental trials prove that, by means of the proposed approach, the Global Capital improves.

Keywords: Cloud federation, Multi-agent System, QoS, Grid Computing, Trust, Decentralized Procedure, Group Formation, Multi-Cloud

1. Introduction

The grid computing paradigm [14] has evolved from the traditional Virtual Organizations (VO) to the federated grid architectures, in which brokers and institutions are able to share resources among different grid infrastructures, thus resulting in a more flexible approach [3]. Such an evolution is strongly due to the increasing complexity of grid tasks [5] submitted by companies and institutions [36], which need to be supported by specialized dynamic VO [13]. Indeed, as grid clients send computational requests characterized by complex requirements, they will benefit from the *collaboration* between grid VOs, which are able to provide specialized resources to the result of the expected computation.

Similarly, in the last years the interest for inter-cloud environments [21] increased. Broker-driven multi-cloud, as well as pure cloud federations offer several advantages to customers: different geographical locations, improved application resilience, avoidance of vendor clock-in [2]. Providers can benefit from this model as they can expand their infrastructures on demand and can offer improved SLA to their customers.

In this context, federated brokers have to deal with complex tasks of resource allocation (i.e. cloud composition [24]), as requirements will involve the evaluation of a huge set of federated resources [7]. Therefore a key issue is represented by the need of achieving a high efficiency in allocating federated resources. In particular, complex task requirements need high priorities, and are strictly required to avoid choices that might cause poor performances. This scenario usually leads to competition among computing nodes, which want to provide the best service to their nodes [28]. On the contrary, it also

implies the presence of possible malicious behaviors, mainly due to service providers which promise performances that will not be actually realized. In such a perspective a suitable trust model [8, 13, 22] can help clients and grid nodes to quantify the expected level of performance and, in general, the level of mutual trust. Existing trust-based approaches have not been specifically designed for federated computing infrastructures and some of them present only strategies for optimal resource allocation without considering trust issues. Basing on the considerations above, in this work we present and discuss a trust-based approach aiming at maximizing the measured "performance" or, in other words, the global Quality of Service (QoS) perceived within a number of federated computing nodes. We focus on the concept of *resource sets*, i.e. the sets of computational resources characterizing complex requirements in federated computing infrastructures. Our solution is based on the use of software agents [15] that, in this approach, manage every node which may be a grid computing element, a grid site, a part of a data center, as discussed in Section 2. Furthermore, the concept of *agent aggregation* (i.e. groups and friendship) is exploited as the basis of collaboration between federated nodes. While the trust model makes it possible to compute some measures of reliability and reputation, an algorithm for agent *Friendship and Group Formation* (FGF) makes it possible to maximize the *Global Capital*, i.e. the "global utility" of the whole federation.

This work is an extended version of a preliminary, abridged study presented in [10], which dealt only with theoretical results. Instead, in this new work, we present a number of experiments showing that the execution of the FGF algorithm – supported by the dissemination of trust information –

allows federated brokers/nodes to improve either individual and global satisfaction.

The plan of the paper is as follows. In Section 2 the reference scenario, as well the role of software agents assisting computing nodes, is described. Section 3 introduces the trust model. Section 4 presents the FGF algorithm designed to lead friendships and groups among agents. In Section 5, an extensive set of experiments is discussed to prove the effectiveness of the presented approach, while Section 6 is devoted to the related literature, which has been analyzed extensively. Finally, in Section 7, we draw our conclusions and introduce our ongoing research.

2. The basic scenario of Federated Computing

In the present approach, the set of all the heterogeneous resources available on the generic federation made by N nodes, let's say $\mathcal{F}$, is modeled as a finite number of R incremental sets of resources, where the R-th set (i.e. the last) includes all the resources available on $\mathcal{F}$. Moreover, the generic service, requiring for its execution the r-th set of resources, will be identified by s^r.

Another important assumption is that the context of the proposed and generic federation may assume a "competitive" character. In other words, whenever the generic client c_j benefits from a service s^r by the node n_i (with $1 \leq r \leq R$), it has to pay a fee p to the provider n_i, whose amount is based on the consumed set of resources r and on the expected quality of the provided service. This assumption can be considered reasonable, especially in the context of multi-cloud, on which broker-driven federations [21] reside between providers and customers in order to negotiate the needed SLA for

various services [21]. Let's assume also that a number of software agents [15], let A be the set of all of them, said *node agents*, assist computational nodes in performing their activities. Formally, each generic agent $a_i \in A$ is associated with the node $n_i \in \mathcal{F}$ and is characterized by a "skill" mapping $\sigma_i(r) \in [0,1] \subseteq \mathbb{R}$, which refers to the measured performance of its services. More in detail, $\sigma_i(r) = 1$ means the maximum quality in providing a service which requires the specific set r of resources and, vice versa, $\sigma_i(r) = 0$ means the minimum quality. Each time a service s^r is provided by the agent a_i (i.e. the node n_i) to the client c_j, a feedback $f \in [0,1] \subseteq \mathbb{R}$ is returned by c_j to a_i, where $f = 1$ means that c_j has perceived the maximum level of satisfaction for s^r, and, vice versa, $f = 0$ will mean no satisfaction. Moreover, we assume 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 $\bigcup_{1<l<k} g_{i_l} \subseteq A$.

One of the key elements of the scenario presented above is the assumption that for each service s^r the agent a_i, that takes charge of the service, can require the *support* of another node n_j (i.e. agent a_j) [28]. If a_j collaborates with a_i by providing the required set of resources and it is a friend of a_i or it belongs to the same group of a_i, this help is provided for free; otherwise, a fee p_s has to be paid from a_i to a_j after such a support has been provided.

To select the best agents to collaborate for the service s^r, a_i can ask a recommendation $rec_j(r)$ about the skill $\sigma_j(r)$ of a_j for a given service s^r to an agent a_k. Also in this case, a_k can accept or refuse the request for $rec_j(r)$ and, due to the adopted competitive scenario, each agent can perform at most Y recommendation requests. If a_k accepts and it is a friend of a_i or it belongs to the same group with a_i, this recommendation will be provided for free;

otherwise, a price p_r has to be paid from a_i to a_k after the recommendation has been provided. However, the final choice is performed by the agent a_i based on the trust model described in Section 3, which, in turn, also considers the *reliability* of the node a_j based on the feedbacks provided by the users that consumed the services.

3. The Trust Model

In the proposed multi-agent context, each agent a_i maintains a triple of values $< \alpha,\ \beta,\ \gamma >$ ranging in $[0, 1] \in \mathbb{R}$ (called respectively *Reliability*, *Honesty* and *Reputation*) for each agent a_j with which a_i interacted in the past.

Reliability. The Reliability of a_j in providing a set r of resources, denoted by $\alpha_{ij}(r)$, represents how much a_i trusts a_j in its capability to provide resources for a service s^r. Furthermore, once a_i received a feedback f for a service s^r, if a_j has contributed to the service s^r, f will include a share $f_j^* \leq f$ due to a_j that will be assigned to it proportionally to its contribution. Thus if a_i completely delegated a_j in providing s^r then the feedback f will be totally of a_j, i.e. $f_j = f = f_j^*$. More formally, the reliability is computed by averaging all the feedbacks received by a_i for all the services that required the set of resources r and the collaboration of a_j.

Honesty. The Honesty of a_j in giving a recommendation to a_i, denoted by $\beta_{i,j}(r)$, represents the overall reliability of the agent a_j in recommending some other agents in providing a set r of resources. It is computed by averaging all the difference between the feedbacks f_{x_1},...,f_{x_s} and the associated recommendation $rec_j{}^{x_1}$,...,$rec_j{}^{x_s}$ received by a_i for some s agents suggested

by a_j.

Reputation. The Reputation denoted by $\gamma_{i,j}(r)$, represents how much, on average, the agents interrogated by a_i provided an estimated value of capability of a_j – in terms of performance referred to the resource set of a_j – which is close to the measured value.

It is computed as the mean of all the recommendations received by a specific agent a_i, about another agent a_j on a resource set r and weighted by the *honesty* of the recommenders.

Three parameters $\overline{\alpha}$, $\overline{\beta}$ and $\overline{\gamma}$ are associate with *Reliability*, *Honesty* and *Reputation*, respectively, as "cold start" values, acting as starting values in case any previous interaction of other agents with a_i has not been performed.

The value of trust computed by a_i about the performance of an agent a_j in terms of resource set r, is denoted by $\tau_{i,j}(r)$, and computed as:

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

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

4. Friendship and Group Formation

Let us assume that, 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 one of its own groups, it will be provided for free (see Section 2). Moreover, in the present scenario we define, for each set of resources r, two sets of *preferred agents*:

- a set ${PC_i}^r$ storing the *preferred contributors* agents with which a_i interacted in the past for a contribution referred to r and that have (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 referred to r and having (i) the Y highest honesty values $\beta(r)$ and (ii) a honesty value greater than the threshold β^{min}.

Therefore, in order to maximize the performance of the services provided by the generic agent a_i, 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:

$$\bigcup_{r \in R} \Big({PC_i}^r \cup {PR_i}^r \Big) = F_i \bigcup \Big(\bigcup_{g \in G_i} g \Big) \tag{2}$$

In order to use a convenient notation, we set

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

therefore Equation 2 can be written as:

$$\bigcup_{r \in R} {PA_i}^r \;=\; F_i \bigcup AG_i \tag{3}$$

Basing on Equation 3, we are interested in measuring two factors: *Loss of performance* and *Additional cost*, as described below.

- **Loss of performance (L).** When some agents belong to the set $\bigcup_{r\in R} {PA_i}^r$ but not to $F_i \bigcup AG_i$, i.e. $\left(\bigcup_{r\in R} {PA_i}^r\right) - \left(F_i \bigcup AG_i\right) \neq \emptyset$, there is a *loss of performance* – with respect to the optimum case, represented by Equation 2 – in providing services if one of those agents is selected. This number can be measured by computing, 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 set 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 define for an agent i a factor, called *Loss of Performance*, in turn composed of two components, $L_i^{(\tau)}$ and $L_i^{(\beta)}$, defined as follows:

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

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 or 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).** In case $\left(F_i \bigcup AG_i\right) - \left(\bigcup_{r\in R} {PA_i}^r\right) \neq \emptyset$, i.e. some agents belong to the set $F_i \bigcup AG_i$ but not to the set $\bigcup_{r\in R} {PA_i}^r$,

we call *Additional Cost* the ratio of agents that will be never contacted by a_i to obtain help for free:

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

We also define the "disadvantage" D_i of a_i, as the average of the sum of the factors $L_i^{(\tau)}$, $L_i^{(\beta)}$ and C_i:

$$D_i = \frac{L_i^{(\tau)} + L_i^{(\beta)} + C_i}{3} \tag{4}$$

Finally, we define the *Global Capital* (GC), by taking into account the whole federation $\mathcal{F}$, as the mean value of all the contributions $(1 - D_i)$ provided by each agent a_i:

$$GC = \frac{\sum_{a_i \in A}(1 - D_i)}{\|A\|}$$

In the following we describe the Friendship and Group Formation algorithm, which is aimed at minimizing (maximizing) the disadvantage D (Global Capital CG).

4.1. The Friendship and Group Formation Algorithm

The Friendship and Group Formation (FGF) algorithm aims at minimizing the disadvantage D_i calculated by Equation 4.

Henceforth we will denote the time elapsing between two consecutive epochs by T (epoch). For each epoch, some preferred agents provide to join with the set $F_i \bigcup AG_i$ in order to replace those agents provided of the worst trust or honesty values. The FGF algorithm consists of two parts:

the former (*Task A*) is periodically executed by each agent a_i to obtain the friendship or the membership in a group of G_i of those agents belonging to the set $\bigcup_{r \in R} {PA_i}^r$ but not yet belonging to the set $F_i \bigcup AG_i$. The second task (*Task B*) consists of a set of subtasks designed to manage the requests of friendship of the other agents and those of joining sent by the other agents to the groups with which a_i is joined or whose it is a leader (administrator).

Task A. The aim of this task is to support the generic agent a_i to obtain the friendship or the membership in a group of G_i of those agents belonging to the set $\bigcup_{r \in R} {PA_i}^r$ but not yet belonging to the set $F_i \bigcup AG_i$. Task A consists of the following ordered sequence of steps:

1. The sets $F_i \bigcup AG_i$, and $\bigcup_{r \in R} {PA_i}^r$ are computed.
2. A friendship request is sent by a_i to each agent $a_j \in \Big(\bigcup_{r \in R} {PA_i}^r - F_i \bigcup AG_i \Big)$.
3. The agent a_j is added to F_i only if it accepts the friendship request. If a_j does not send the answer within a given time threshold, a timeout is triggered such that the request is considered rejected.
4. If the friendship request is refused by a_j, then a_i executes the steps:
   (a) a_i request from the DF the set G_j of all the groups having a_j as member (described in Section 2).
   (b) a_i computes the disadvantage D_i^* for each group $g \in G_j$.
   (c) For all the group $g \in G_j$ such that $D_i^* < D_i$ and D_i^* is minimum a joint request is sent to them. If the group g does not give an answer within a fixed time threshold, the request is considered rejected.

(d) If g accepts the membership request, then this group is added to G_i, otherwise a_j is added to the set C_i.

5. If C_i is not empty, then a *call for a new group* is sent to all the agents belonging to it by a_i. In presence of some agents which agree to constitute a new group, then such a group is formed and registered into the DF.
6. When an agent a_j is added to the set F_i, then the worst friend agent a_k will be removed from F_i. More in detail, the agent a_k is selected as follows:
   - if $a_j \in PC_i^r$, then the agent $a_k \notin \left(\bigcup_{r \in R} {PA_i}^r \right)$ having the worst trust value $\tau_{i,k}(r)$ is selected or
   - if $a_j \in PR_i^r$, then the agent $a_k \notin \left(\bigcup_{r \in R} {PA_i}^r \right)$ having the worst honesty value $\beta_{i,k}(r)$ is selected.

Task B. The second task consists of three subtasks designed to manage the requests for friendship of the other agents and those of joining sent by the other agents to the groups with which a_i is joined with or whose it 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 arrives at an agent a_i coming from an agent a_j, then:
  1. a_i computes a new value for the disadvantage ${D_i}^*$ by adding the agent a_j to the set F_i and, at the same time, it removes an agent a_k as described by Step 5 of Task A, Step 1).

2. a_i will accept the request coming from a_j only if ${D_i}^* \leq D_i$; otherwise, this request will be refused.

- *Membership Request.* When the administrator of a group g receives the Membership request coming from an agent a_j, then it promotes a vote (positive or negative) among all the agents belonging to g. The proposal can be accepted by a majority. In particular, each agent a_k will send a consensus only if the insertion of a_j in its same group g will not increase its disadvantage D_k.

- *Call for a new group.* Such a request is sent from an agent a_j and it is accepted by a_i only if the insertion of a_j in the set $F_i \bigcup AG_i$ does not increase its disadvantage D_i.

We observe that operations that can lead software agent to enter a waiting state, which may potentially cause a distributed deadlock, are accompanied by a simple timeout mechanism (see points 3 and 4 of Task A). Basically, time thresholds should be fixed, also for recommendations coming from recommenders, which does not change the semantic of the algorithm. Therefore, as specified in Section 4.1, it easy to observe (by constructing the wait-for graph) that deadlock cannot occur in the provided algorithm.

If we take into account the subsequent resource allocation in the considered federated environments, we can say that the distributed algorithm will give a result (group and friendship organization) to each agent in finite time. In addition, the proposed model does not provide specific mechanisms for resource allocation (e.g. mechanisms concerning transactions for distributed resources), as it only gives suggestions for selection of resource sets, which are

based on the proposed trust metrics and the execution of the FGF algorithm. Basing on the consideration above, it can be observed that no deadlock for resource waiting will occur deriving from the dynamics introduced by the proposed multi agent systems.

5. Experiments

In this Section we present some experimental results in order to show the practical implications given by the application of the trust and the aggregation model presented in this paper.

5.1. Simulated scenario and related parameters

The results presented in this Section have been obtained by a set of simulations performed by means of the software Octave [20] and are based on a set of parameters summarized in Table 1.

The scenario of the proposed solution is composed by a number of federated nodes that hold the same set of resources, i.e. they are able to provide the same services, but they show, on average, different behaviors in terms of reliability, as reported in table 1.

As stated in [11], the reliability of a service or, in other words, the probability of failures, generally depends on several factors which can be grouped into three main classes: (*i*) failures or losses of performance due to the request layer; (*ii*) resource management layer; (*iii*) network, program and physical resource layer. As a consequence, the final reliability if any, is the result of the combination (i.e. the product of the different reliabilities) of factors' groups as classified above. As the trust model presented in Section 3 provides a single value for measuring the reliability of a federated node, we prefer not

Table 1: Simulation parameters.

Simulation Parameter	**Value**
N_{nodes}	1000
LP nodes / MP nodes / HP nodes	30% / 40% / 30%
L.P. feedbacks values (normal distr.)	$\{\mu, \sigma\} = \{0.4, 0.4\}$
M.P. feedbacks values (normal distr.)	$\{\mu, \sigma\} = \{0.7, 0.2\}$
H.P. feedbacks values (normal distr.)	$\{\mu, \sigma\} = \{0.9, 0.1\}$
Recommendations (range of gen. values)	$[\tau - 0.1\tau, \tau + 0.1\tau]$
No. of feedbacks (Poiss. distr.)	$\lambda = 50$
No. of recommendations (Poiss. distr.)	$\lambda = 20$
Nodes degree distribution, power law	$\{C, \alpha\}$=$\{14.42, 2.50\}$

to explicit the different factors affecting the reliability. Therefore, we simulated a single index of reliability for each node. It represents the overall QoS provided by the single node.

In the discussed model feedbacks reflect measured reliability (i.e. QoS) of the services, therefore we simulated the nodes reliability by generating different values for the feedbacks (see Section 3) by sampling from a normal distribution with different mean and standard deviation [26]. To this end, we grouped the nodes into three sets based on their performances, i.e. high (H.P.), medium (M.P.) and low (L.P.). Table 1 shows the different values of μ and σ for the three categories of nodes. A plot of the correspondent *probability density function* (p.d.f.) is depicted in Figure 1. We also supposed that the number of services (i.e. generated feedbacks) per step and the number of recommendations are independent events. Therefore, we generated feedbacks

and recommendations by means of the Poisson distribution [26] with parameter $\lambda = 50$ (mean) for the feedbacks and $\lambda = 20$ for the recommendations, as indicated into Table 1 and Figure 2.

We generated an initial network of agents/nodes by adopting the power law model [9, 31]. In this model, the network is generated by setting the degree of the nodes (i.e. its probability distribution) being compliant with a power function $P(x) = Cx^{-\alpha}$. We set $\alpha = 2.5$, which refers to that commonly accepted (i.e. measured) for the social networks [9], and $C = 14.4278$ [31]. In addition, to normalize the underlying area, we truncated the function at $X = X_{min} = 5$, which is the minimum degree of the generated network.

The resulting probability distribution is shown in Figure 3.

In order to study the effects of the proposed model (and algorithm) we used three different simulation profiles, identified as **R**, **T** and **T+FGF**:

1. **R**: nodes select their collaborators randomly.
2. **T**: nodes are supported by the trust system, proposed in Section 3, in selecting their collaborators. The experimental results are discussed in

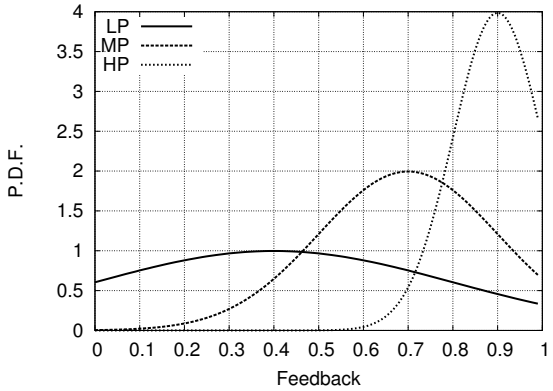

Figure 1: Different values of feedbacks simulated for different nodes.

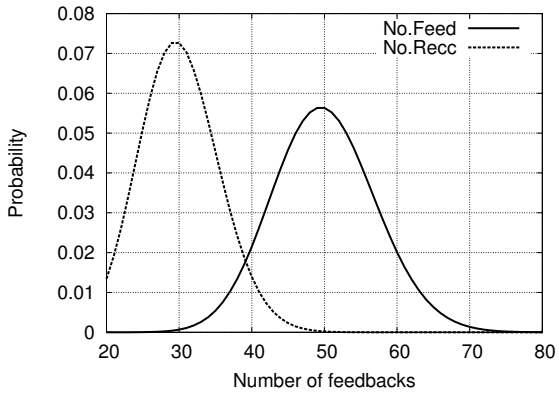

Figure 2: Probability distribution for the number of generated feedbacks and recommendations per step (Poisson distribution).

Section 5.2, along with those obtained in the previous point, for direct comparison.

3. **T+FGF**: nodes are supported by the trust system, proposed in Section 3, in selecting their collaborators. In addition, they execute the FGF algorithm, discussed in Section 4. The experimental results are in Section 5.3.

The results shown in Section 5.2 and Section 5.3 have been obtained by averaging 100 simulations having the same set of parameters.

5.2. Random vs trust-based selection

As stated before, the first set of results has been obtained in the case resource sets are selected randomly (case **R**). Therefore, in this case there is no trust system to support the collaboration between nodes. In particular, the selection is performed only by means of the availability, of a specific

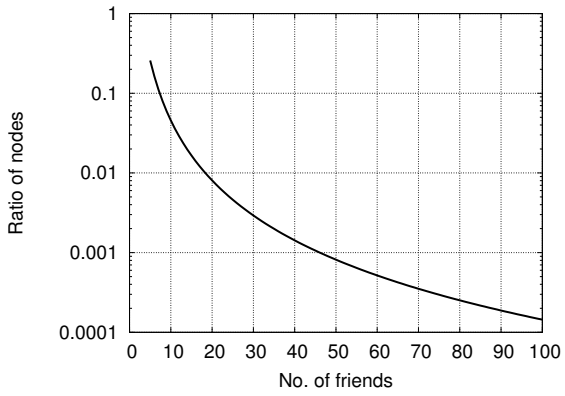

Figure 3: Nodes degree of the initial network.

node, to provide a specific service. The second set of results, (labeled **T** in Section 5.1), has been obtained with the support of the trust model described in Section 3. In detail, by assuming that a_i has to select a node to start a collaboration for a service, we set its behavior as follows:

- If a_i doesn't hold trust information, any suitable collaborator within the federation can be selected.
- a_i can ask (and will always receive) a recommendation from one or more of its friends.
- To select a node suitable for a service by means of trust information, all the nodes trusted by a_i in the range $[\tau_H - \omega_\tau \tau_H, \tau_H]$ will be taken into account, where τ_H is the higher trust index among the nodes for which trust information are available, and ω_τ ranges in $[0, 1] \in \mathbb{R}$.

Figure 4 shows a comparison of the results obtained for random (**R**) and trust-based (**T**) selection. For the latter, we set $\omega_\tau = 0.1$ and $\delta_i = 0.5$

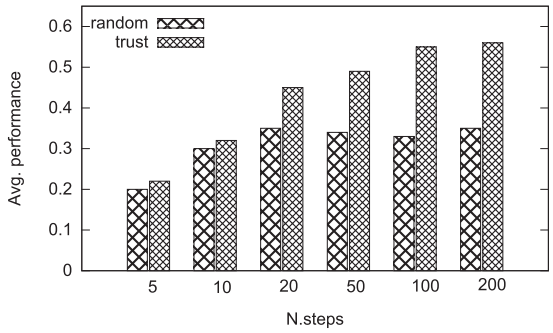

Figure 4: Average feedback values of the "top" (most active) 20 nodes.

(see Section 3, Equation 1). The values in the y-axis (Avg. performance) represents the average of the feedbacks related to all the collaborators of the 20 most active nodes, i.e. the set of 20 nodes which contacted the largest number of collaborators. Data plotted in Figure 4 makes evidence that, after a brief transition, the (positive) effects of the trust system come out causing a rise in the average level of performance for the nodes of the federation involved in the selection of suitable collaborators.

Figure 5 summarizes an additional number of results obtained with trust-based selection only, 200 steps and $\delta_i = 0.5$. In this case, we set different values of ω_H, which ranged from 0.1 to 0.5. We can observe that when the threshold has a value of about 0.5, the trust-based selection behaves in a way similar to the random-based selection, which represents the expected behavior. Moreover, a value between 0.1 and 0.2 will give the best result in terms of average performance when selecting collaborators for services.

5.3. Trust-based selection and execution of FGF algorithm

In this set of simulations, the trust system has been coupled with the execution of the FGF algorithm discussed in Section 4.

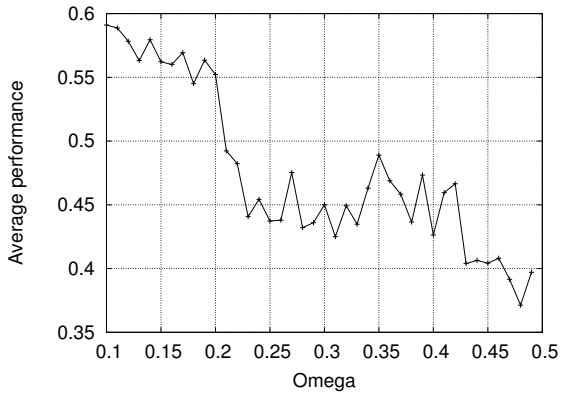

Figure 5: Trust based selection for different values of ω_τ .

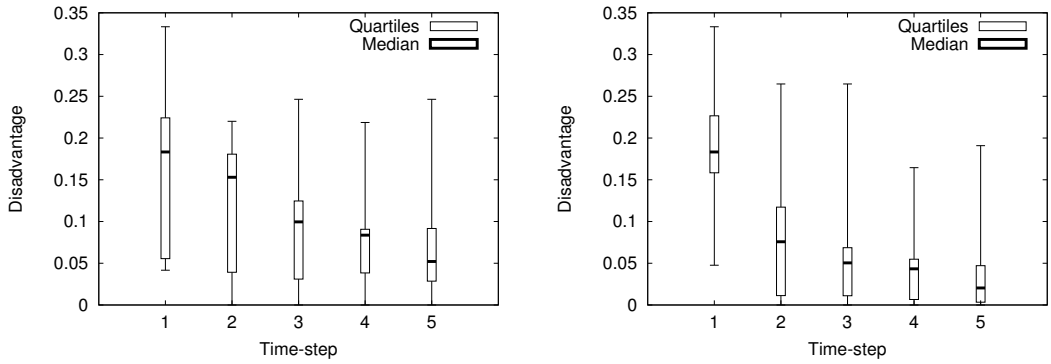

Figure 6: Disadvantage (D). Min, 1st Quartile, Median, 3th quartile, Max. $\tau^{min} = \beta^{min} = 0.2$. Figure 6a (left): X=Y=10, Figure 6b (right): X=Y=40.

Measure of Disadvantage (D). First, we report some results obtained from a set of experiments aimed at confirming that the execution of the FGF algorithm will give a contribution by lowering the Disadvantage D or, in the same manner, by rising the average *Global Capital* $GC = \frac{\sum_{a_i \in A}(1 - D_i)}{\|A\|}$. The simulation parameters are those described at the beginning of the current section Table 1. In this set of experiments, the collaborators for services were

selected on the basis of the trust system, as in the experiments described in Section 5.2. Results are shown in Figures 6-7. We report only the first 5 steps of the simulation as the trend stabilizes very quickly with a very low average disadvantage.

Candlesticks shown into Figure 6 provide the median, quartiles and outliers of the Disadvantage D 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 sets PC and PR, as previously discussed in Section 4. While Figure 6a summarizes the results for $X = Y = 10$, Figure 6b refers to $X = Y = 40$, where X and Y are the maximum size of the sets PC (Preferred Contributors) and PR (Preferred Recommenders). We observe that the median value of the disadvantage has a downward trend. Moreover, as the sets PC and PR grow in size from $X = Y = 10$ (Figure 6a) to $X = Y = 40$ (Figure 6b), the median (so did quartiles) assumes lower values very quickly, which is the expected behavior.

Results shown in Figures 7a and 7b report the median value of the Disadvantage D, for X and Y ranging from 10 to 40 by steps of 10. Moreover, in Figure 7a we set $\tau^{min} = \beta^{min} = 0.2$, while in Figure 7b we set $\tau^{min} = \beta^{min} = 0.5$. By comparing data plotted in Figure 7a with those of Figure 7b, we can observe that the more selective is the parameter τ^{min} (resp. β^{min}), which is the minimum value of trust (resp. honesty) to put a node into the set PC (resp. PR), the greater will be, on average, the disadvantage.

Execution of the FGF algorithm. The last set of simulations represents an attempt to measure the improvements due to the execution of the FGF algorithm. To this purpose, we repeated the same set of simulations by

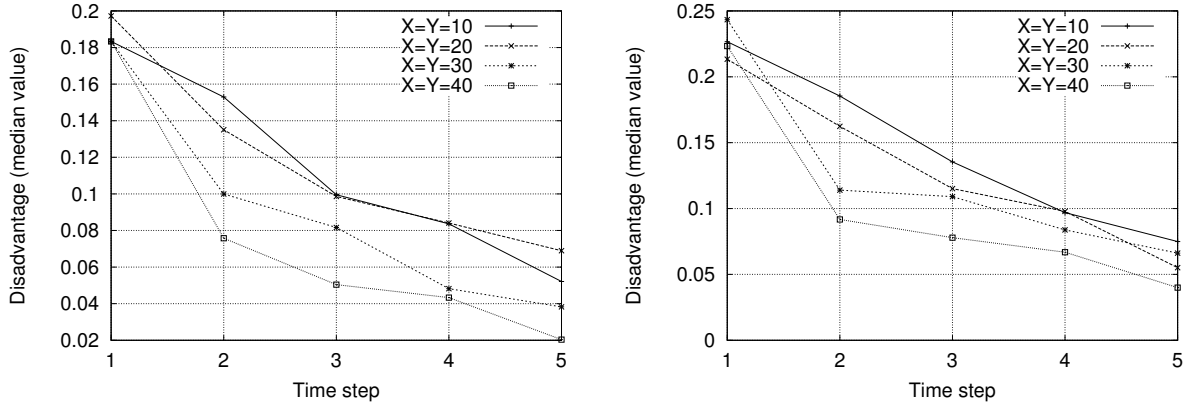

Figure 7: Median value of disadvantage. Figure 7a (left): $\tau^{min} = \beta^{min} = 0.2$. Figure 7b (right): $\tau^{min} = \beta^{min} = 0.5$

setting the FGF parameters $\tau^{min} = \beta^{min} = 0.2$, $X = Y = 40$, and $\omega_\tau = 0.1$. The first choice is due to the fact that, as discussed above and shown in Figure 7a, the lower the values of those parameters, the lower the index of disadvantage. Differently from the previous set of simulations, we let the nodes select their collaborators both from the list of friends and from the groups they belong to. In Figure 8, on which we compare the performances related to the trust-based selection, already shown in Figure 4, with those obtained by the new set of simulations, which includes the execution of the FGF algorithm. In this last set of results we can observe that there is a performance improvement over the results shown in Section 5.2, in the order of $15 - 20\%$. This behavior is fairly intuitive. Indeed, as specified in Section 4, the nodes always try to connect to the set of Preferred Contributors (best nodes in terms of performance) plus that of Preferred Recommenders, which will result in better opportunity to improve the provided QoS.

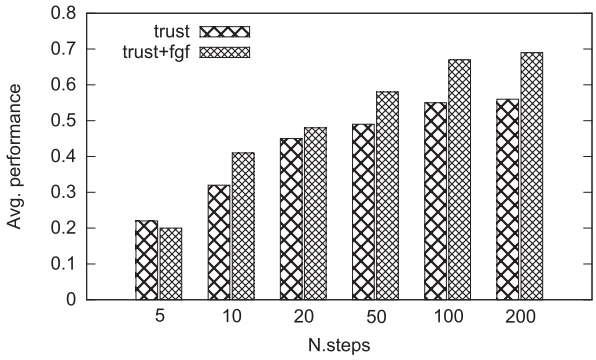

Figure 8: Trust-based selection vs trust-based+FGF

5.4. Comparison with other approaches

We compared our method with three approaches concerning group formation and trust evaluation. The first considered approach is that discussed in [6], on which clusters of agents are generated by considering global trust in a competitive scenario. This approach – named *CLTRUST* in the following of this section –, trades on the trust measure to suggest the best agents to contact as fruitful interlocutors. It does not deal with the issue of improving the global capital of the agent community, as in our approach, which is based on a meritocracy criterion. Nevertheless, it is interesting to measure the global capital resulting from the application of this approach within the experimental scenario described in the previous section, where cluster formation is driven by the global reputation measure. In particular, global reputation of each agent a_k in *CLTRUST* is computed as $\frac{1}{m}\sum_{i=1}^{m}\tau_{ik}$, as described in [6], while values of trust used in our approach – say *FGF*

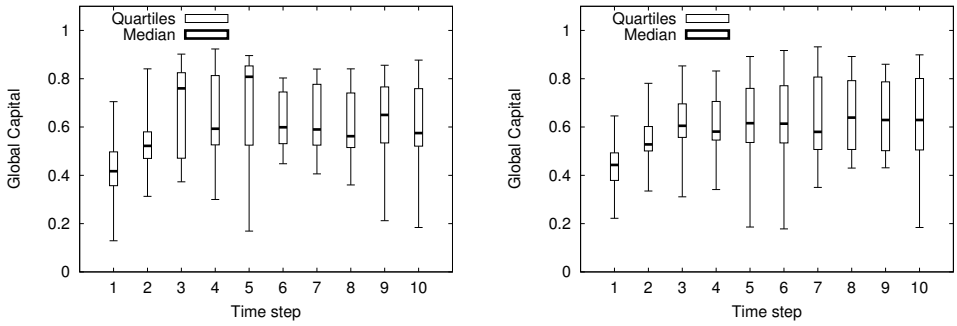

Figure 9: Global Capital. Five number summary. Left: KMNS (kmeans). Right: KMED (kmedoids). X=Y=10

are computed locally, as described in Section 3. Two well known clustering approaches were also considered: k-means [23] (*KMNS*) and k-medoids [25] (*KMED*). Cluster detection was executed at each simulation step for each different approach. Moreover, as *KMNS* and *KMED* are used to generate clusters of agents based on a number of features, we used global reputation plus an additional feature which codifies the set of resources of the node itself. The ratio behind this choice is that resulting clusters include agents that has similar values of reputations and a set of resources. Among the parameters used for the simulation, that have been discussed in the previous subsection, the algorithm was executed until the value of Global Capital appeared stable enough. Moreover, we set $X = Y = 10$, where X and Y are the number of prefered contributors (PC) and recommenders (PR), respectively.

For each experiment, we collected the five number summary of the Global Capital, which are reported in Figures 9-10. In particular, Figure 9-left shows the results for *KMNS*, Figure 9-right shows the results for *KMED*, while Figure 10-left shows the results for *CLTRUST*, and Figure 10-right shows the

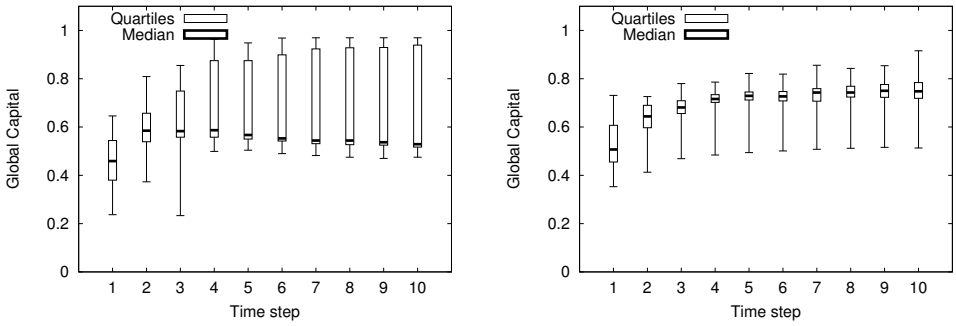

Figure 10: Global Capital. Five number summary. Left: CLTRUST. Right: FGF. X=Y=10

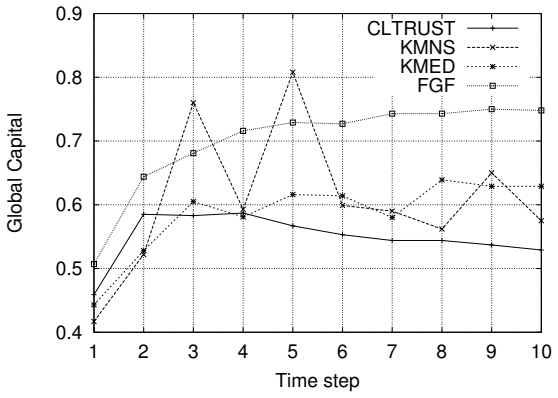

Figure 11: Median value of Global Capital. X=Y=10

results for our approach, *FGF*. Interestingly enough, although *CLTRUST* makes use of the global reputation as measure to clusterize agents, it does not perform well as *KMNS* and *KMED*, that have, on average, similar performance. Nevertheless, our approach (*FGF*) outperforms its competitors, on average, of about 20%. This is more noticeable through Figure 11, which shows the trend of the four compared approaches in terms of median values. Moreover, we can observe that *FGF* has the shortest IQR (Inter-Quartile

T-step	*CLTRUST*	*KMNS*	*KMED*	*FGF*
1	0.46	0.42	0.44	0.51
2	0.58	0.52	0.53	0.64
3	0.58	0.76	0.60	0.68
4	0.59	0.59	0.58	0.72
5	0.57	0.81	0.62	0.73
6	0.55	0.60	0.61	0.73
7	0.54	0.59	0.58	0.74
8	0.54	0.56	0.64	0.74
9	0.54	0.65	0.63	0.75
10	0.53	0.57	0.63	0.75

Table 2: Global Capital (median) – Time-steps 1-10 – CLTRUST, KMNS, KMEd, FGF

Range) (Figure 10-right), and a very stable trend (Figure 11). Data shown in Figure 11 is also reported in Table 2, on which each column represents the median value of the Global Capital with the other four approaches.

5.5. Discussion

By the results presented in Section 5.2 we show that, for a given network of agents assisting a number of federated computing nodes with different level of reliability, relying on the distributed FGF algorithm supported by the trust model presented in Section 3, will give a significant advantage in terms of performances.

The second part of experiments, reported in Section 5.3, shows that the index of Disadvantage D (or, at the same manner, the Global Capital GC)

will improve if the FGF algorithm is being executed in the network. We also observed that the computed average disadvantage of the nodes approaches to a stable configuration more or less quickly, depending on some parameters characterizing the aggregation model (FGF algorithm) discussed into Section 4. Analogously, we verified that the performances due to the trust-based selection can be improved by adopting the FGF algorithm, which is the expected behavior. Indeed, by means of the execution of the FGF algorithm, friendship is built basing on the trust level of the nodes and the agents will aggregate also in groups by reflecting the trust level and the honesty in providing a recommendation. In particular, by joining groups, agents are able to keep the reference of a number of agents able to help them to select reliable collaborators, event they don't accept their friendship request. We also found a $15 - 20\%$ improvement of performances, (i.e. measured reliability) when the FGF algorithm was executed, is in the order of $15 - 20\%$, which can be considered significant.

6. Related Work

This section discusses an extensive set of research work related to the issue of partner/node selection in the context of self-interested agents and grid systems. We survey the literature related to these topics, with particular emphasis to the proposal tied to our work, and focus on the principal metrics proposed to deal with the problem addressed in this work.

Many of such models are based on direct observations and/or on communications with other agents. Moreover, they consider different criteria as trust, reputation, provided QoS, etc.

In the information exchange domain, research on belief revision also involves how to select appropriate information providers. Belief is, in general, a situational awareness, and research investigating belief revision in multi-agent systems [1, 4, 16, 35] pursues a similar objective: build the agents' beliefs accurately and efficiently by using all the information provided.

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. Coalition formation seeks to partition the agents in a system into groups which maximize the utility of the group or the individual agent. The partitioning of the agents is usually modeled as a characteristic function game and involves three activities [34]: *(i)* coalition structure generation, *(ii)* solving the optimization problem of each coalition, and *(iii)* pay-off division. The first two activities are closely related to find appropriate partnerships from a set of potential groupings, while pay-off division is to decide how the utility gained by forming a coalition should be distributed among the agents to keep the coalition stable.

Recently, some proposals adopted trust in competitive agent systems [19, 32], for instance, to constitute clusters of agents [6, 18] and for generating recommendations in social network contexts [12] or to detect group of actors in a competitive social community [27, 29, 33]. These approaches trade on trust measures to suggest the best agents to contact as fruitful interlocutors, but none of them deal with the issue to improve the social capital of the agent community on the basis of a meritocracy criterion. Differently, our proposal introduces a meritocratic principle in order to obtain such an advantage, also by encouraging the actors to assume correct behaviors in order to improve

their reputation.

As discussed in Section 5.4 we compared our approach with two classical clustering approaches, namely k-means [23] and k-medoids [25], which can be used to group agents. Nevertheless, the approach presented in this paper is based on a distributed algorithm which organizes friendships and groups, by which the Global Capital improves significantly.

Authors of [17] combined the principles and the concepts found in social networks to design a decentralized and adaptive resource discovery approach in complex grid systems. Experimental results show how the relationship between clusters can improve the resource discovery processes and allow different resource distributions and user request patterns to a better adaptation for. However, this approach lacks of a component meant to enhance the social capital of the agent (node) community by improving meritocracy.

We recently considered in [13] a specific model of Grid Federation aimed to improve the QoS in dynamic grid federations by focusing on the role of *Grid Virtual Organizations* (VO). To this purpose, we have proposed to group VOs into large-scale federations on which the original goals and scheduling mechanisms are left unchanged, while grid nodes can be quickly instructed to join with or leave any VO at any time basing on the measurements of past behaviors in terms of costs and performances. This introduces a relevant flexibility for resource management, and users can benefit from this flexibility in terms of QoS.

Another approach we have recently proposed, is related to federated computing infrastructures [30]. In this case, a trust model was integrated into an P2P overlay network which enable fully decentralized, efficient finding

approach.

7. Conclusions and future work

The main purpose of a federation of computing infrastructures is to exploit both potential *collaboration* and *competition* between providers. In both the cases, brokers have to deal with the selection of an optimal set of resources.

Therefore, a key issue is represented by the necessity of achieving a high efficiency in allocating federated resources, by taking into accounts that complex job require high priorities and it is necessary to avoid choices which might cause unbalanced resources allocations.

In this context, we introduced a partnership-based model to optimize the global QoS of a number of federated resources. In the proposed model computational nodes are supported by intelligent agents, which manage friendships and group memberships. Furthermore, (*i*) computational *resource sets* support tasks in the federation, (*ii*) *agent aggregation* (i.e. friendships and group memberships) are the basis of collaboration among 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.

A specific distributed algorithm, called Friendship and Group Formation (FGF), allows federated nodes to select their partners (friends and group memberships) to improve the global QoS. To 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 federated nodes; the (*ii*) *Global Capital* (GC), a global index,

telling us how well the Brokers/Nodes of the Federations can work together when a computational task requires an inter-site/nodes collaboration.

The validity of the proposed model is supported by an extensive set of experiments. Results clarly show that the adoption of the FGF algorithm, suitably supported by the proposed trust model, the Global Capital (which reflect the global QoS) of the Global Federation is effectively improved.

In our ongoing research, we plan to better study the influence of several parameters characterizing our model. Furthermore, we plan to restrict the scope of the work to pure federation, in order to study the effects the FGF algorithm under a number of additional hypothesis.

ACKNOWLEDGEMENTS

This work has been partially supported by (i) PRISMA PON04a2 A/F funded by the Italian Ministry of Education, University, and Research; (ii) Program Programma Operativo Nazionale Ricerca e Competitivitá 2007-2013, project BA2Kno (Business Analytics to Know) PON03PE 00001 1, in Laboratorio in Rete di Service Innovation; (iii) Program Programma Operativo Nazionale Ricerca e Competitivitá 2007-2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research; (iv) Networks and Complex Systems (NeCS) Laboratory of the Department DICEAM - University Mediterranea of Reggio Calabria.

References

[1] C.E. Alchourron, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symbolic*

Logic, 50(2):510–530, 1985.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Comm. of the ACM*, 53(4):50–58, 2010.

[3] M. Bandieramonte, A. Di Stefano, and G. Morana. An ACO inspired strategy to improve jobs scheduling in a grid environment. In *Algorithms and Architectures for Parallel Processing*, pages 30–41. Springer, 2008.

[4] K.S. Barber and J. Kim. Soft security: Isolating unreliable agents from society. In *Trust, Reputation, and Security: Theories and Practice*, pages 224–233. Springer, 2003.

[5] B. Boghosian, P. Coveney, S. Dong, L. Finn, S. Jha, G. Karniadakis, and N. Karonis. Nektar, spice and vortonics: using federated grids for large scale scientific applications. *Cluster Computing*, 10(3):351–364, 2007.

[6] F. Buccafurri, A. Comi, G. Lax, and D. Rosaci. A trust-based approach to clustering agents on the basis of their expertise. In *Agent and Multi-Agent Systems: Technologies and Applications*, pages 47–56. ACM, 2014.

[7] R. Buyya and R. Ranjan. Special section: Federated resource management in grid and cloud computing systems. *Future Generation Computer Systems*, 26(8):1189–1191, 2010.

[8] L. Chunlin and L. Layuan. Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Applied Intelligence*, 25(2):147–158, 2006.

[9] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[10] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné. A qos-aware, trust-based aggregation model for grid federations. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*, volume 8841 of *LNCS*, pages 277–294. Springer Berlin Heidelberg, 2014.

[11] Yuan-Shun Dai, Yi Pan, and Xukai Zou. A hierarchical modeling and analysis for grid service reliability. *Computers, IEEE Trans. on*, 56(5):681–691, May 2007.

[12] P. De Meo, A. De Meo, D. Rosaci, and D. Ursino. Recommendation of reliable users, social networks and high-quality resources in a social internetworking system. *AI Communications*, 24(1):29–50, 2011.

[13] P. De Meo, F. Messina, D. Rosaci, and G. M. L. Sarné. An agent-oriented, trust-aware approach to improve the qos in dynamic grid federations. *Concurrency and Computation: Practice and Experience*, 2015.

[14] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. of High Performance Computing Applications*, 15(3):200–222, 2001.

[15] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent agents III Agent Theories, Architectures, and Languages*, pages 21–35. Springer, 1997.

[16] K.K. Fullam and K.S. Barber. Using policies for information valuation

to justify beliefs. In *Proc. of the 3rd Int. Joint Conf. on Autonomous Agents and Multiagent Systems-Vol. 1*, pages 404–411. IEEE, 2004.

[17] L. Gao, Y.S. Ding, and H. Ying. An adaptive social network-inspired approach to resource discovery for the complex grid systems. *International Journal of General Systems*, 35(3):347–360, 2006.

[18] S. Garruzzo and D. Rosaci. Agent clustering based on semantic negotiation. *ACM Trans. on Autonomous and Adaptive Systems*, 3(2), 2008.

[19] S. Garruzzo, D. Rosaci, and G. M. L. Sarné. Integrating trust measures in multi-agent systems. *International Journal of Intelligent Systems*, 27(1):1–15, 2012.

[20] GNU Octave. http://www.gnu.org/software/octave/.

[21] Nikolay Grozev and Rajkumar Buyya. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, 44(3):369–390, 2014.

[22] L.L.Z. Guo-So, Yuan and J.L.L.J. Chang-Jun. Dynamic level scheduling based on trust model in grid computing. *Chinese J. of Computers*, 7:023, 2006.

[23] John A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

[24] Amin Jula, Elankovan Sundararajan, and Zalinda Othman. Cloud computing service composition: A systematic literature review. *Expert Systems with Applications*, 41(8):3809–3824, 2014.

[25] Leonard Kaufman and Peter J Rousseeuw. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis*, pages 68–125, 1990.

[26] Paruchuri R Krishnaiah. *Handbook of statistics*. Motilal Banarsidass Pub., 1980.

[27] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné. A distributed agent-based approach for supporting group formation in P2P e-learning. In *13rd Int. Conf. of the A.I.A.I., Proc. of*, pages 312–323. Springer, 2013.

[28] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné. A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures. *Future Generation Computer Systems*, 2015.

[29] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G.M.L. Sarné. Hyson: A distributed agent-based protocol for group formation in online social networks. In *Multiagent System Technologies*, pages 320–333. Springer, 2013.

[30] F Messina, G Pappalardo, D Rosaci, C Santoro, and GML Sarné. A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures. *Future Generation Computer Systems*, 56:77–94, 2016.

[31] Mark EJ Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[32] D. Rosaci. Trust measures for competitive agents. *Knowledge-based Systems (KBS)*, 28(46):38–46, 2012.

[33] D. Rosaci and G. M. L. Sarné. Matching users with groups in social networks. In *Intelligent Distributed Computing VII*, pages 45–54. Springer, 2013.

[34] T.W. Sandhlom and V.R.T Lesser. Coalitions among computationally bounded agents. *Artificial intelligence*, 94(1):99–137, 1997.

[35] Glenn R Shafer and Prakash P Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2(1-4):327–351, 1990.

[36] J. Yu and R. Buyya. A novel architecture for realizing grid workflow using tuple spaces. In *Grid Computing, 2004. Proc. 5th IEEE/ACM Int. Work. on*, pages 119–128. IEEE, 2004.