This is a postprint version of the paper: P. De Meo, F. Messina, D. Rosaci, G.M.L. Sarnè. An agent-oriented, trust-aware approach to improve the QoS in dynamic Grid Federations. Concurrency and Computation: Practice and Experience (**CCPE**). Volume 27, Issue 17. Pages 5411 -5435.
The final version is available at DOI: 10.1002/cpe.3604. 2015. Wiley.

An agent-oriented, trust-aware approach to improve the QoS in dynamic Grid Federations

Pasquale De Meo[1], Fabrizio Messina[2*], Domenico Rosaci [3], Giuseppe M. L. Sarné [4]

[1]*DICAM, University of Messina, 98166 Messina, Italy*
[2]*DMI, University of Catania, Viale A. Doria 6, Catania, I-95125 Italy*
[3] *DIIES University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Calabria, Italy*
[4] *DICEAM University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Calabria, Italy*

SUMMARY

In this paper a distributed approach aimed at improving the Quality of Service in dynamic grid federations is presented. Virtual Organizations (VO) are grouped into large-scale Federations on which the original goals and scheduling mechanisms are left unchanged, while Grid nodes can be quickly instructed to join or leave any VO at any time. Moreover, an agent-oriented framework is designed to observe and characterize past behaviors of nodes in terms of resource sharing and consumption, as well as to determine the trust relationships occurring between each pair of nodes. By combining trust and historical behaviors into a unified convenience measure, software agents are able to evaluate the *i*) advantages of node's membership with VOs, and *ii*) whether a specific set of nodes is able to meet the actual requirements, in terms of resource sharing and consumption of a specific VO. The convenience measure has been exploited to design a fully decentralized, greedy procedure, aimed at controlling the Grid Formation process. Extensive simulations have shown that the coordinated and decentralized process of Grid Formation provides a powerful mean to improve the overall Quality of Service of the Grid Federation.

Received . . .

KEY WORDS: Grid Computing, Grid Federation, Grid Formation, Multi-agent System, Quality of Service, Trust

1. INTRODUCTION

The paradigm of Grid Computing gained a large popularity in a broad range of application contexts and it is now widely acknowledged as a standard to efficiently share computational resources [1], especially for scientific projects [2]. As institutions and companies have joined many ad hoc Grid Virtual Organizations (VOs) [3, 4, 5], a great interest emerged to study and propose how these grids can collaborate for sharing their resources and to mutually coordinate for solving complex tasks.

By means of *Grid Federations* multiple and heterogeneous grids are coupled together [6, 7, 8, 9, 10]. Research on Grid Federations is now at a mature stage and it catches the interest of large communities of researchers coming from different domains like Climate [11], Science [12] and Finance [13]. Grid Federations can be modeled in various ways, as for instance by means of a fully decentralized approach for resource allocation which does not involve any broker of coordination [14]. In other words, each organization simply shares its clusters at large scale via a peer level coupling.

*Correspondence to: Fabrizio Messina, Department of Mathematics and Informatics, University of Catania, Viale A. Doria 6, Catania I-95125, Italy. E-mail: messina@dmi.unict.it

The original approach for constituting Grid VOs is that of "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions" [15] for a *common objective*, i.e. Virtual Organizations are strongly constructed around some specific objectives. Our view of interoperability in Grid Federations strictly adheres the original conception of Grid Computing. Therefore our view of Grid Federation is that of leaving unchanged original VO policies, such that the achievement of common goals remains central, by providing *policies, algorithms and necessary automation* to allow grid nodes to join with or leave automatically any VO in the federation. On the other hand, each grid of the federation should be able to accept or refuse the joining request of a node autonomously.

The construction of such a very dynamic scenario is really encouraged by the strong development of technologies for utility computing, mainly virtualization [16, 17] and software for creating Public and private Clouds[18], by which quick reconfigurations and deploying of virtual nodes with specific middleware and software are efficiently supported. In principle, this approach can meet various demand changes, such that the overall Quality of Service (QoS)[19, 20, 21] provided within the Virtual Organization can remain stable and the original aims and autonomy of the VO remain unchanged. We refer to such a scenario, for convenience, as *Open Grid Federation* ($\mathcal{G}$).

In the scenario described above, we can state some simple questions: given a grid node n_i (e.g. a Research Department or a firm division) and a grid $g_j \in \mathcal{G}$, how can we objectively measure the advantage that n_i would gain when joining a grid g_j? Conversely, how can we decide if it is convenient that n_i leaves a grid to which it is already joined? More formally, in order to provide an answer to these questions, the problem can be formulated as "*dynamically assigning nodes to grids with the goal of optimizing the QoS offered by the grids*". We will refer to this problem, for convenience, as the *Grid Formation*.

When dealing with the issue above, i.e. Grid Formation, we consider several aspects. First of all, we take into account the main characteristic of a Grid node, not only in terms of resources, but also in terms of requests coming from grid users. Indeed, grid users belonging to a particular grid node will exploit its own access pattern, in terms of nature and quantity of requested resources, and such a pattern depends on the goal to be achieved by the node itself when joining with that particular Grid. From this point of view, balancing supply and offer of resources can help to avoid unsatisfied requests and unallocated resources. Since the nature and quantity of resources each grid node can offer or ask on behalf of its own grid user is highly variable, predicting it is almost impossible. Anyway, by observing jobs and service requests submitted to the node in terms of amount and nature of involved resources, a behavioral profile for the grid node can be suitably constructed.

Another concern we take into account is the reliability of grid nodes, since *not all the interactions* in a grid systems can be satisfactory for all the involved parties. To this end, grid nodes should be able to rate the reliability of other nodes in automatic manner, on behalf of their own grid users. Such a rating should encode how much a node is satisfied by its past interactions with another node which has provided services on the basis of a Service Level Agreement (SLA) specified by its user. Therefore, *feedbacks* describing the level of QoS provided by the nodes should be properly collected and aggregated to determine the *trustworthiness degree* of a target node, which should be also weighted by means of the relevance of the provided services.

Given the premises above, in this paper we propose a distributed approach to help grid nodes in performing an automatic selection of Grid VOs to join with, and to help a VO administrator to select the nodes to accept into his own VO. In this approach we take into account the *past node behaviors* and the *trustworthiness* of a node. In particular, we consider the behavior of each node *i*) in terms of costs of the resources requested/offered and *ii*) the trust of a node that depends by the relevance and the feedbacks of the provided services, the number of performed interactions and their freshness. The combination of trust and historical behaviors is aimed at introducing the *convenience* measure, which represents the advantage – for a given grid node – to join with a certain grid and vice versa. On the basis of this convenience measure, we designed a simple and fully decentralized procedure, with greedy characteristics, called Grid Formation (GF), in order to associate nodes with grids and vice versa. The aim of the GF procedure is to associate nodes with grids by means of a suitable matching problem which makes use of the convenience measure in a distributed fashion.

DOI: 10.1002/cpe

The trust model discussed in this work, which leads in the definition of an integrated trust measure, is specifically designed to take into account reliability and reputation criteria, which are weighted by means of the knowledge that each node acquired in the past about its provider, and the freshness of such a knowledge. Furthermore, differently from other trust systems, our approach is designed to derive some information about the relevance and the level of QoS of the services provided by nodes, such that it is particularly suitable to be used in a grid environment. In such a way, with respect to the provided grid services, we derive homogeneous evaluations which do not suffer of the possible different opinions or insight of each agent about the QoS of a given service [22]. The resulting trust system is eventually able to recognize malicious behaviors in a few steps, as shown by the experimental results discussed in Section 6.

The key contributions of this work — which started with a preliminary study already presented in [23] and discussed at the end of Section 7 — can be summarized as follows:

- We introduce a computational framework designed to capture past behaviors of grid nodes (i.e. to obtain behavioral measures), on behalf of their users, as well as to compute the trust between a pair of nodes.
- We design a trust model specially conceived for a grid context and to be used with a behavioral measure to obtain a unique and integrated measure (i.e. the convenience measure).
- We introduce a multi-agent architecture for managing nodes, grids and their interactions, along with the features of the GF algorithm, a fully decentralized procedure designed to exploit the convenience measure in order to re(organize) the composition of each grid of the federation.
- We provide a set of experimental results concerning the simulation of the trust model in a simulated Grid Federation. These have been obtained by considering some critical scenarios, such that the *resilience* of the trust system to malicious activities.
- We present a further set of experimental results obtained in a simulated Grid Federation of growing complexity – i.e. up to 16,000 nodes assigned to up to 40 grids – which have shown that the GF procedure is able to compute the convenience measures of nodes and grids in few iterations with an advantage in terms of accuracy of about 40% with respect to a merely random assignment of nodes to grids.

The plan of the paper is as follows. Section 2 reports some preliminary definitions and assumptions on the involved scenario, as well as the Grid Formation problem in an Open Grid Federation, along with the behavioral and trust measures. In section 3 we describe a reference multi-agent architecture to support the execution of the GF decentralized procedure, which is discussed into section 4. Then, in Section 5 we discuss some possible practical applications of the proposed approach by means of a realistic case study. In Section 6 we present and analyze the experimental results, while in Section 7 we compare our approach with some related works. Finally, in Section 8 we draw some final conclusions.

2. MODEL

2.1. Preliminary Definitions and Assumptions

Our experience with grid systems refers to a number of simulations performed on the grid nodes of the Italian Grid Infrastructure [24], built with Glite middleware [25][†], the product of a European project aimed at supporting European Grid infrastructures.

A simplified schema of a Grid Virtual Organization is depicted in Figure 1. User authentication is done by single sign-on and delegation [15] through the *User Interfaces* (UI) provided by their own

[†] At the time of writing it is part of the EMI project [2]

DOI: 10.1002/cpe

institutions. Grid users submit, monitor and control computational jobs through the UI available in their own institutions. VO members are Academic departments, research institutes, companies, and so on, having interests in sharing their own resources within a VO. Resources are organized into logical computational entities called *Computing Element* (CE) and *Storage Element* (SE), in order to allow grid users to access to a set of heterogeneous resources to employers, scientists and so on.

Grid jobs are described by means of *Job Description Language* (JDL), and are submitted to the *Workload Management System* (WMS) [26], which can denote as "broker". Its responsibility is to select suitable computing elements basing on the matching between requirements contained into the JDL (e.g. by selecting short or long queues) and the resources characteristics. It is interfaced with CEs, which in turns are made by a *Local Resource Manager* (LRM), e.g. the Torque [27] scheduler, which manages a set of Worker Nodes (WN - not shown in Figure 1), i.e. physical of virtual CPUs. Similarly SEs are built by means of management software which is called SRM (Storage Resource Manager).

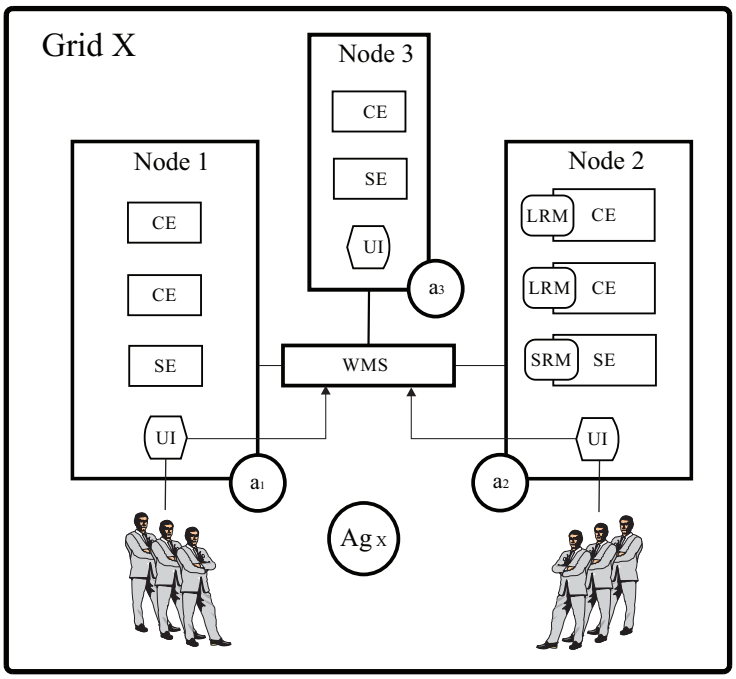

Figure 1. A simplified architecture of a Grid Virtual Organization

Computing elements provide a wide range of services, depending on the capabilities of the hardware and the software made available (numerical routines, software libraries, and so on) by local administrators. Code for user computations may reside *i*) on the worker nodes, ready for execution, or can be *ii*) compiled into worker nodes along with the needed libraries. In the second case binaries may be left into SEs along with some libraries which do not reside in the worker nodes due to administration choices. After the binaries have been compiled on the worker node, they can be left on the SE for subsequent retrieval and execution. Indeed, we used the second approach (*ii*) to run a set of simulations through ComplexSim [28].

In our model we suppose that a grid *node* is composed by a number of CEs and SEs, and at least a User Interface, as depicted in Figure 1. Furthermore, in the following we use the generic term *service* to indicate the occupancy of resources due to the execution of a number of jobs which are related to the same user, e.g. part of a workflow [29], or bags-of-tasks [30].

2.2. Technical concerns about reliability on Grid Computing

As Section 2.5 contains the description of the trust models adopted in our approach, in this section we explain what reliability is in our approach.

DOI: 10.1002/cpe

As stated in Section 2.1, our experience with Italian Grid Infrastructure is mainly represented by the submission of bags-of-tasks for simulation through ComplexSim [28]. When interfacing with the WMS through the UI (see Fig.1) we used a number of hand-made scripts to check the status of the jobs, retrieve and collect the related results. Since we submitted many bags-of-tasks – where each task was a single job representing a simulation – each time a job was faulty, the "agent" running in the UI (i.e. the set of scripts) recovered information about failed jobs and (re)submit to the WMS.

Type of faults	Example	I.	A.
Problems with proxy	Proxy expired, problem in detecting the lifetime of the proxy	√	√
Problems with scheduler	Scheduler misconfiguration, errors in file staging	√	X
Problems operating with DATA	Problems connecting with SE, problem with the File Catalog	√	X
Problems with network	Networks timeout between CE and WN	√	X
Authorization	Misconfiguration on the WN, grants on DB	√	X
Incorrect/lack of maintenance on the CE	Excessive Size of Log files	√	X
Resources not available	Incorrect requirements, problem with CE	√	√

Table I. Errors

There are several technical reasons for which jobs may abort when submitted to Grid WMS and eventually to the CE. We may refer to the possible faults as *infrastructural* – i.e. due to some components of the grid – and *applications* related to. A sample of possible faults is reported in Table I. The last two columns indicate whether the fault is due to the infrastructure (I.) or to the application (A.). For instance, the first error reported in the table – "Problem with Proxy" – is marked as both infrastructural and related to the application. Indeed, the first case may happen when the CE has wrongly detected the lifetime of the proxy, while in the second case the proxy has expired, in others words the user did not renewed it before submitting the job. As a further example, we can look at the second to the last row, which is related to oversized log files, that is clearly a case of fault due to an infrastructural reason. Another example is reported in the last row, which represents the case on which the specified resource (e.g. a queue for long jobs managed by the scheduler of a specific CE) is not available. In this case the error may be due to the fact that the Grid Information System [31] may give incorrect information (infrastructural) or that requirements specified in the JDL [25] were incorrect.

In the following we assume that the reliability of grid nodes is evaluated basing on infrastructural behaviors, i.e. the faults strictly related to the the user code are not taken into account.

2.3. Behavioral and trust measures in Open Grid Federations

Suppose $\mathcal{N}$ be the space of n grid nodes, and $\mathcal{G}$ the set of Grid VOs which include the nodes in $\mathcal{N}$, i.e. the Grid Federation. In the Grid Federation model conceived in this approach, VO policies (e.g. security, resource sharing, scheduling) are left unchanged. Therefore, each computational grid can provide a set of computational services according to the objectives and the identity of the underlying VO, maintaining its own autonomy. On the other hand, in order to get the opportunities arising from the connection of several grids, suitable *policies and automation* should be provided to allow grid nodes to automatically join to or leave any VO of the federation and, similarly, each grid of the Federation can decide to accept or refuse the joining request of a node. This assumption can be considered realistic due to the massive use of virtualization [16, 17, 32], by which nodes can be quickly reconfigured to become part of a different VO.

Furthermore, we assume that decisional processes aimed at joining to or leaving a grid are supported, on each node $n_i \in \mathcal{N}$, by a software agent, let be a_i. Similarly, we can assume the "reasoning capability" of each federated grid is implemented by another software agent A_j [33]. Nodes and Grid Agents are depicted into Figure 1.

DOI: 10.1002/cpe

In this section we present a set of measures aimed at studying *i)* the nodes behaviors, in terms of offered and required resources, and *ii*) the overall grid behavior, which is computed on the basis of the behavior of the nodes forming the VO. In particular, three different measures are defined: *i*) *behavioral* measures, *ii*) *trust* measures and *iii*) *convenience* measures, as described in the following of this Section.

2.4. The Behavioral Measures

Each grid node is also characterized by the community of Grid users submitting job requests for Grid services through its UI (User Interface), e.g. scientists of a research institute which joined a VO to perform a set of specific simulations. Grid users submit requests which have to be assigned to the computational resources of the grid through the VO resource broker (WMS in Figure 1). The set of resources requested by the grid users of a specific node will be denoted as "required" resources. Vice versa, offered resources are those shared by the specific node within the VO.

The definition of the behavioral measures relies on the consideration that grid users characterized by a relevant or specific need of resources are interested in joining to VOs which hold suitable capabilities. Similarly, Virtual Organizations lack of resources which are particularly requested by their own users, have the interests to include other nodes holding those resources.

In order to model the issue above we define the *Resource Cost* associated with the node k (i.e. RC_k) as the actual cost of a service with respect to the amount of resources offered/required by k:

$$RC = \sum_{i=1}^{k} r_i \cdot c_i \tag{1}$$

where r_i is the i-th offered/required resource expressed in resource units (for instance, MIPS for CPU, TB for storage, etc.) and c_i is its associated unitary cost.

We also define the "historical" attitude of the generic node n_k to offer or require resources within a grid by adopting the *Node Behavior* (NB) measure computed as:

$$NB_k^{(t)} = \alpha \cdot NB_k^{(t-1)} + (1-\alpha) \cdot \frac{RC_{k,req}^{(t)}}{RC_{k,req}^{(t)} + RC_{k,off}^{(t)}} \tag{2}$$

where $RC_{k,off}$ is the cost of offered resources, while $RC_{k,req}$ is the cost of requested resources. The resources requested by the node n_k is the amount of resources requested/consumed by those users accessing the VO of n_k through the UI of n_k (see Fig. 1). The value of $\alpha \in [0,1] \subset \mathbb{R}$ determines the relevance of these two contributions. In particular, the new value of $NB \in [0,1] \subset \mathbb{R}$ at the time t is computed by weighting the previous value (i.e. computed at time $t-1$) by the parameter α, and by considering the contribution due to the new service (i.e. computed at time t) for which the node has been involved as provider and/or consumer weighted by $(1-\alpha)$.

In detail, the second contribution is calculated between the cost of the involved requested and offered resources, R_{req} and R_{off} respectively. When $RC_{req} >> RC_{off}$, then the contribution results to be $NB \approx 1$ and it means that its own user tends to ask more resources to other nodes of its own grid. Vice versa $RC_{req} << RC_{off}$ means $NB \approx 0$, therefore the node is generally self-sufficient and/or it tends to offer its own resources to the other nodes.

Since we are also interested in the evaluation of the *footprint* of a grid in offering or consuming resources, we define the *Grid Behavior* (GB) as the average of the behavioral measures NB_k for all the nodes n_k that joined to the specific grid g_j. More formally:

$$GB_j^{(t)} = \frac{1}{||g_i||} \sum_{k=1}^{||g_i||} NB_k^{(t)} \tag{3}$$

Clearly $GB_j \in [0,1] \subset \mathbb{R}$ represents the tendency of the whole grid g_j to offer or require resources.

DOI: 10.1002/cpe

2.5. The Trust Measures

The second measure we introduce in our framework is based on the concept of *trust*. In our scope, we can assume a common definition of trust: *the quantified belief by a truster with respect to the competence, honesty, security and dependability of a trustee within a specified context* [34]. In the model described in this work, the truster is the generic node n_p (i.e. the associated agent a_p), on the behalf of its own grid user, with respect to the generic node n_r (i.e. the associated agent a_r). For this aim, we can refer again the Figure 1, remembering that the technical concerns about reliability of grid nodes have been discussed in Section 2.2. Furthermore, the notions of *reliability* and *reputation* used in this work are detailed below.

The reliability is a measure of perceived trust derived by the direct knowledge due to the *interactions* occurred in the past. Differently, reputation can be assumed as an expected behavior based on information (i.e. indirect knowledge) on the past interactions occurred with other counterparts [35]. In other words, the reputation of a node can be considered as a measure of the trust perceived by the whole community of agents and calculated by exploiting their opinions due to their past interactions occurred with that node. So that when the direct knowledge is not enough to rely on a node, then the reputation measures become essential for evaluating trust.

In order to make clear the meaning of *interactions* in the proposed model, we can state that an interaction between two nodes n_p and n_r is represented by the process by which the resource broker allocate jobs belonging to n_p users into resources of node n_r. Software agents acting on behalf of node n_p can observe jobs submitted by the users and query the resource broker on behalf of them to store information from which it is possible to extract reliability indexes. For instance, let's assume that a bag of tasks [36, 37] is submitted by grid users of node n_p and a part of it has been allocates into node n_r. Then the agent a_p analyzes the behavior of node n_r by monitoring the status of the jobs, running time, and so on, in order to extract some reliability index useful to compute the reliability of node n_r (see Section 2.1).

Let $\tau_{p,r}$, $\eta_{p,r}$ and $\rho_{p,r}$ be respectively the measures of trust, reliability and reputation that the generic node n_p (i.e. agent a_p) computes for the node n_r (i.e. agent a_r). The trust measure $\tau_{p,r}$ is obtained by combining the reliability ($\eta_{p,r}$) and the reputation ($\rho_{p,r}$) measures suitably weighted by the real coefficient $\beta_{p,r}$, ranging in $[0, 1] \in \mathbb{R}$, in the form:

$$\tau_{p,r} = \begin{cases} 0.5 & t = 0 \\ \beta_{p,r} \cdot \eta_{p,r} + (1 - \beta_{p,r}) \cdot \rho_{p,r} & t > 0 \end{cases} \tag{4}$$

Note that for new coming nodes the initial trust (i.e. reputation) is set to 0.5 in order to penalize them for not too much [38] but enough to contrast whitewashing strategies [39].

The coefficient $\beta_{p,r}$ increases according to the number of interactions occurred between the nodes and their freshness, because also the direct knowledge that n_p has of n_r should improve.

More formally, the coefficient $\beta_{p,r}$ is computed as:

$$\beta_{p,r} = \frac{I_{p,r}}{I_{max}} \tag{5}$$

where I_{max} is a system threshold representing the number of interactions after which the "knowledge" that a node has about another node is maximum. Moreover, $I_{p,r}$ is incremented or decremented at each new step as described in Equation 6.

$$I^t_{p,r} = \begin{cases} max(1, I^{(t-1)} - 1) & \text{if } \Delta T_{p,r} > \Delta T \\ min(I_{max}, I^{(t-1)} + 1) & \text{if } \Delta T_{p,r} \leq \Delta T \end{cases} \tag{6}$$

where ΔT is a system threshold representing the maximum time interval between the current time-step (t) and the time at which the last interaction ($i_{p,r}$) occurred between node n_r and n_p, i.e. $\Delta T = t - T_{i_{p,r}}$. Therefore, the ratio adopted in computing $\beta_{p,r}$ is to provide a different relevance to the reputation with respect to the reliability based on the experience acquired by n_p about n_r and the "freshness" of such an experience. In other words, the contribute of the reputation in computing trust decreases as much as the number of the interactions occurred between the two involved nodes constantly increases. Therefore, whenever the last interaction with n_r is older than

DOI: 10.1002/cpe

ΔT, the reputation increases in relevance, i.e., the reliability decreases in relevance (first row of Equation 6). Vice versa, whenever the last interaction with n_r is fresh enough (the last interaction with n_r is not older than ΔT), the relevance of reliability will increase.

A fine tuning of parameters ΔT and I_{max} will allow to assign *i*) different relevance of the reliability and the reputation and to increment the *ii*) resilience to unexpected behaviors changes.

2.5.1. Computation of reliability. The reliability measure, $\eta_{p,r} \in [0,1] \subset \mathbb{R}$, is computed as:

$$\eta_{p,r}^{(t)} = \vartheta_{p,r} \cdot \sigma_{p,r} + (1 - \vartheta_{p,r}) \cdot \eta_{p,r}^{(t-1)} \tag{7}$$

where the parameter $\vartheta_{p,r}$ weights in a complementary way the contributes due to *i*) the feedback parameter $\sigma_{p,r} \in [0,1] \in \mathbb{R}$ computed by n_p with respect to the service $s_{p,r}$ provided by n_r at time-step t and *ii*) the value of $\eta_{p,r}$ computed at time-step $(t-1)$. The parameter $\vartheta_{p,r}$ is calculated basing on the relevance measure (ψ) assigned by n_p to the current service $s_{p,r}$ and the average of the relevance measures of all services provided by n_r to n_p in the past, as follows (Eq. 8):

$$\vartheta_{p,r} = 0.5 - \frac{\psi_{p,r} - \overline{\psi}_{p,r}}{2} \tag{8}$$

where $\psi, \vartheta \in [0,1] \in \mathbb{R}$ and $\overline{\psi}_{p,r}$ is the average value of $\psi_{p,r}$ computed over all the services to which node r provided a contribution. In particular, the form of the Eq. 8 has been adopted to limit the phenomenon by which some nodes could acquire positive feedbacks by means of services assuming marginal importance, e.g. simple jobs submitted in that nodes for testing purposes.

2.5.2. Computation of Reputation. The reputation measure $\rho_{p,r}$ is computed by a node n_p with respect to a given node n_r as a value ranging in $[0,1] \subset \mathbb{R}$. Through the usual meaning of these indexes, 0 means that n_r is totally unreliable, while 1 means that node n_r is totally reliable.

Since each node (i.e. agent) has only a partial view of its community, the trust measures computed in its own might differ from those computed by including the opinions of the whole community. In particular, when the node n_p is interested to calculate the reputation of the node n_r it can ask an opinion about n_r to the node (agent) n_q. We assume the agent a_q, associated with n_q, will provide its opinion to a_p consisting of its own trust measure about n_r (i.e. $\tau_{q,r}$).

Since reputation represents an *indirect measure*, evaluating the capability of a node of providing reliable opinions about other nodes represents an important task. To this purpose, we introduce the *confidence* factor ($\omega_{p,q}$) to weight such an opinion by taking into account the concordance between the trust value that n_p and n_q have about n_r. More formally, $\omega_{p,q}$ is computed as:

$$\omega_{p,q}^{(t)} = 1 - \frac{|\tau_{p,r} - \tau_{q,r}|}{2} \tag{9}$$

Consequently, the reputation $\rho_{p,r}$ is computed by a_p as:

$$\rho_{p,r} = \frac{\sum_{q=1}^{||R_{p,r}||} (\omega_{p,q} \cdot \tau_{q,r})}{\sum_{q=1}^{||R_{p,r}||} \omega_{p,q}} \tag{10}$$

where $||R_{p,r}||$ is the cardinality of the set of agents which provided an opinion about n_r .

Looking at Eq. 10, the confidence factor minimizes the impact of untrustworthy opinions by providing more relevance to those mentors that a_p evaluates as the most similar, like to real life.

DOI: 10.1002/cpe

2.6. The convenience measures

In order to measure the *convenience* for a node n_i to join with the grid g_j we define $\gamma_{i,j}$ as follows:

$$\gamma_{i,j} = |NB_i - GB_j| \cdot \frac{\sum_{n_k \in g_j} \tau_{i,k}}{||g_j||} \tag{11}$$

where $||g_j||$ is the number of nodes affiliate with g_j. In words, the convenience $\gamma_{i,j}$ for n_i to join with g_j increases with the difference between the behaviors of n_i and g_j, and with the average trust computed by n_i of all the nodes belonging to g_j.

Similarly, in order to measure the *convenience* for a grid g_j to accept the request of a node n_i to join with g_j itself, we define $\eta_{j,i}$ as follows:

$$\eta_{j,i} = |NB_i - GB_j| \cdot \frac{\sum_{n_k \in g_j} \tau_{k,i}}{||g_j||} \tag{12}$$

The measure $\eta_{j,i}$ takes into account the behaviors of n_i and g_j, on the one hand, and the trust computed by all the node $n_k \in g_j$ for n_i .

Note that, in general, $\eta_{i,j} \neq \eta_{j,i}$, i.e. the trust computed by the node n_i to the node n_j is not the same of that computed by n_j for n_i, therefore Eq. 11 and 12 assume different values for the nodes n_i and n_j .

3. THE MULTI-AGENT ARCHITECTURE SUPPORTING OPEN GRID FEDERATIONS

In our model, grid nodes and VO administrators are supported by intelligent software agents [33] capable of performing all the activities aimed at (re)organizing the VOs based on the measures presented in the Section above.

We assume also that a Directory Facilitator (DF) agent is associated with the whole agent framework, and will be able to provide a yellow page service to all the other agents.

The multi-agent architecture described in this section is synthetically represented in Figure 2. In the right part a grid federation of 3 VOs is depicted. For each of them, a Grid Agent is indicated to manage the global information associated to the grid and to take the decisions (as discussed below in this Section), while in the left part we depicted some other nodes. In order to not complicate the draw, the possible overlapping of VO membership is not considered.

3.1. Agent profiles

Each agent is able to build, manage and update its own profile, i.e. the knowledge needed to execute its own activities. The profile of an agent a_i associated with a node n_i can be denoted by means of the tuple $P_i = \langle WD_i, RD_i, BD_i \rangle$, where:

- WD_i (*Working Data*): it is the set of necessary endpoints to communicate with the DF (i.e. to contact the other agents), to interact with the resource manager of the node n_i and the User Interface which offers to the users the access to the grid.

- RD_i (*Resources Data*): it is the set of data concerning all the resources *used*/*offered* in the past by the node and the agents which made available additional resources and/or *consumed* resources of this node.

- BD_i (*Behavioral Data*): it is the set of data concerning the behavioral measures of node n_i and grid agents to which the agent has joined.

DOI: 10.1002/cpe

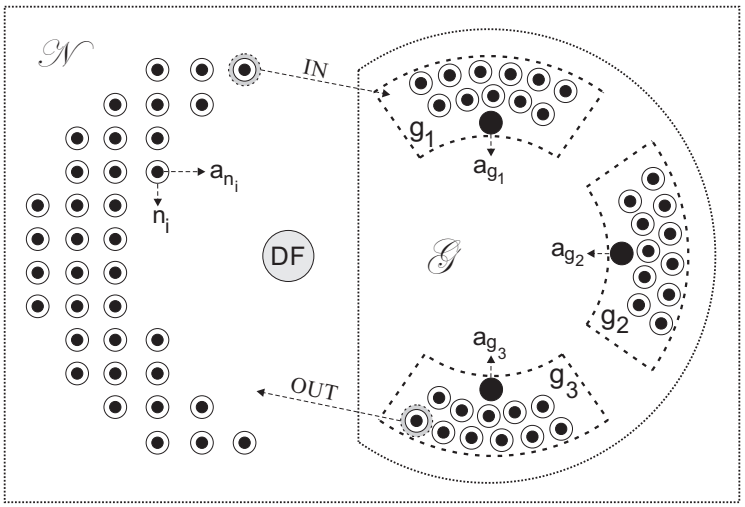

Figure 2. The multi-agent architecture for the Open Grid Federation.

3.2. Agent Behaviors.

Node and grid agents execute a set of tasks in coordination, which is briefly summarized here.

The Node Agent Behavior. The behavior of a node agent consists of:

- updating behavioral measures whenever its own users have consumed resources of other nodes or, vice versa, jobs of users belonging to different nodes have used its own resources;
- updating its trust measures about the other nodes of the federation, based on the service requests performed by the grid users belonging from its own node;
- periodically updating its own convenience measures, as described in Section 2.5;
- sending behavioral and trust measures to the software agents representing its own grids, whenever it has changed;
- receiving behavioral measures from the grid agents of the Grids it has joined to;
- asking to any grid agent of joining to the associated grid, based on the convenience measure;
- sending a leaving message to the grid agent of one of the grids it joined with in the past, based on the convenience measure.

The Grid Agent Behavior. On the other hand, the behavior of a Grid Agent is composed by a few steps which are coupled with the correspondent steps of the node agents behaviors. More in particular, it consists of:

- receiving the behavioral and trust measures from the agents of the nodes within its own Grid;
- updating the overall behavioral measure of its own Grid whenever an update for a behavioral measure has been received from a node of its own Grid;
- periodically updating the convenience measure of its own Grid, based on the behavioral and trust information received from the nodes of its own Grid;
- asking to a node agent of joining to or leaving the VO it represents and manages, due to an updating of the convenience measure with respect to all the grid nodes of the VO;

DOI: 10.1002/cpe

- evaluating a joining request coming from nodes of other grids;
- sending the behavioral measure of the VO to the node agents belonging to its own Grid.

The activities described above, performed by node and grid agents, are formalized in a procedure called GF, i.e. Grid Formation, in Section 4.

4. THE DECENTRALIZED PROCEDURE FOR GRID FORMATION (GF)

In this Section we describe a decentralized procedure named GF (Grid Formation), which is composed by some activities based on the measures defined in Section 2. The activities of the GF procedure differ for node agents and for grid agents, respectively.

To this purpose, let T be the time between two consecutive executions (steps) of the GF procedure. Moreover, we assume that agents can query a distributed database named GR (Grid Repository) on which the list of the grids of the open federation is maintained.

4.1. The GF procedure performed by the node agents.

Let X_n be the set of the grids which the node agent a_n is affiliated to, where $||X_n|| \leq N_{MAX}$, and N_{MAX} is the maximum number of grids which a node agent can join with. We suppose that a_n stores into a cache the grid profile p_g of each grid contacted in the past (see Section 3) and the timestamp d of the execution of the GF procedure for that grid. Finally, let ξ_n (a timestamp) and $\chi_n \in [0,1] \subset \mathbb{R}$ be a threshold fixed by the agent a_n .

The GF procedure performed by the node agent a_n is represented by the following, ordered steps:

I - A set Y of N_{max} grids from GR, so that $X_n \bigcap Y = \{0\}$ is randomly selected.

II - For each grid $g \in Z = (X_n \bigcup Y)$ such that $d_g > \xi_n$, a message to the agent a_g for asking its profile p_g is sent.

III - For each received p_g, the convenience measure $\gamma_{n,g}$ (see Section 2.6) between the profiles of the node agent a_n and the grid g is computed.

IV - The list L_{good}, with all the grids $g \in Z$ such that $\gamma_{n,g} > \chi_n$, is updated.

V - A second list L'_{good} is built by inserting a number $k = min(N_{max}, ||L_{good}||)$ grids of L_{good} with the greater value of $\gamma_{n,g}$.

VI - For each grid $g \in L'_{good}$, if $g \notin X_n$, a join request to a_g together with the profile p_n is sent.

VII - For each $g \in X$, $g \notin L'_{good}$, then the agent a_n deletes its node n from g, i.e. a message to a_g in order to leave the grid g is sent.

4.2. The GF procedure performed on the grid agent side.

Let K_g be the set of the nodes affiliated to the grid g, where $||K_g|| \leq K_{MAX}$, being K_{MAX} the maximum number of nodes permitted by the administrator of the grid. Suppose that the grid agent a_g stores into its cache the profile p_u of each node $u \in K_g$ and the timestamp d_u of its acquisition. Moreover, let ω_g (a timestamp) and $\pi_g \in [0,1] \subset \mathbb{R}$ be thresholds fixed by the agent a_g.

The tasks comprising into the GF procedure and performed by the grid agent a_g, are triggered whenever a join request by a node agent a_n (along with its profile p_n) is received by a_g, as follows:

I - For each node $u \in K_g$, such that $d_u > \omega_g$, a message is sent to the node agent a_u to require the profile p_u associated with u.

II - The convenience measure $\eta_{g,u}$ (see Section 2.6) for each node $u \in K_g \bigcup \{n\}$ – where n is the node which has asked to join with the grid g – and the profile of the grid g is computed.

DOI: 10.1002/cpe

III - The list of good candidates K_{good} storing all the nodes u such that $\eta_{g,u} > \pi_g$ is updated.

IV - A second list K'_{good} stores a number $s = min(K_{max}, ||K_{good}||)$ of nodes belonging to K_{good} having the greater value of $\eta_{g,u}$.

V - For each node $u \in K_g$, if $u \notin K'_{good}$, the grid agent a_g deletes u from g and notifies u. Clearly, if $n \in K'_{good}$, its request to join with g is accepted.

5. CASE STUDY

Partners joining VOs share a number of computational resources which are generally different in nature and quantity. In particular, as grid nodes belong to different administrative domains, they show, in general, different behaviors because they are managed differently. Even Grid users, i.e. scientists and employees which make use of Grid services, will make a different use of resources.

The case study discussed in this section is represented by a distributed and computational intensive application, aimed at studying the behavior of a P2P overlay network for resource finding and allocation [40, 41, 42, 43]. The application consisted of a high number of simulations (i.e. bags of tasks) performed by means of the simulator ComplexSim [28]. The grid resources made available for its execution belonged to the Grid VO Cometa [44], which brought together computational resources across several local Universities and scientific institutions.

First of all, we briefly discuss the characteristics of jobs submitted to the VO, in order to discuss the impact of reliability on the services provided to the grid users. For this aim, it is useful to show a number of directives (i.e. part of the configuration) taken by the simulator (see Listing 1). As shown in Listing 1, some keys take numerical values (e.g. number and degree of nodes), as well some parts (e.g. resource finding) which take pre-defined values. As a consequence, by combining different values for several different parameters, we generated *thousands of configurations* for simulations. Furthermore, about 50-100 jobs were submitted in correspondence of each configuration, such that average values were computed from the results obtained from each configuration. We simulated very large networks, from thousands to millions nodes (parameter *nodes*), by obtaining simulation times of about 1-8 hours, depending on the number of nodes and activities involved into simulations. As a result, we submitted a high number of bags-of-tasks to the queues arranged for "long" jobs[‡]. The jobs were distributed by the WMS (see Figure 1) into the Worker Nodes of the VO, based only on the requirement about the expected duration of the jobs (long jobs queue was selected).

Reliability. One of the relevant issues was that we collected *various errors related to the proxy and file staging* (see Table I), which caused the failure of various jobs that, in turns, had to be resubmitted to the WMS. In this scenario, if a suitable software agent capable of profiling the reliability of the other nodes – and capable of receiving/send recommendations from/to its own peers – were available, a detailed profile of the grid VO could be constructed. Furthermore, by looking at the third part of the configuration file (see Listing 1), it can be observed that the simulator is instructed to produce various traces. In average, this led in generating several MB of data per simulation for subsequent analysis. Furthermore, the analysis of results and log files represented a further step on which, in spite of the fact that elaboration is simple, a good reliability in the communication between SEs and CEs was crucial (cf. Table I). In particular, certain grid nodes appeared to be unreliable, as we collected communication errors between CE, SE and file catalog (cf. Table I). We refer the reader to the discussion about the performance of the trust system, reported into Section 6.1.

[‡]I.E. for jobs with execution times not greater than 12 hours.

DOI: 10.1002/cpe

Listing 1: Template configuration for simulations

```
####overlay network #####
nodes=100
min_deg=2
max_deg=4
nres=2

####JOBS####
mean_jobs=0
jobgen_max_v=200
jobgen_min_v=50
#jobgen_prob_distribution=(random_uniform|poisson|steady)
#job_starting_point=(ls|ms|r) less_suitable/most_suitable/random_start
max_job_dur=10
min_job_dur=5

####LOGGING/TRACING####
trace_node_degree=0
trace_nodes=0
trace_average_short_path_length=0
trace_advances_per_request=3
job_trace=job_trace1.txt
freq_ratio_graph_data_trace=-1
freq_ratio_mass_center_trace=1.1

####RESOURCE FINDING#####
res_finding_alloc_metric=(worst_fit|best_fit|first_fit)
res_finding_lm_strategy=(jump|backtracking|none)
res_finding_forward_strategy=(worst_fit|best_fit|first_fit|max_con|mass_center
    |best_fitBD|mass_centerBD)
res_finding_fwd_strategy=first_fit
#Rf: history check (exclude_already_visited|include_already_visited)
res_finding_fwd_history_check=exclude_already_visited
```

Grid resources. Another important concern is related to the nature and quantity of resources shared from each partner, which, in our case, had a significant impact on our work. Indeed, in order to run our jobs we needed a great amount of powerful CPUs, and our VO dedicated about a half of the total CPUs on long term jobs. On the other hand, we would have had benefits from accessing to the grid infrastructure built within the SCOPE [45] project, which includes a lot of powerful CPUs. Unfortunately, that infrastructure was not part of the VO Cometa which, as stated before, were used for the simulations, and interoperability – e.g. making simulations on those nodes and leaving results on the SE of VO Cometa – was not possible. In this case, by supposing to compute the convenience measures described in Section 2.6 and subsequently the GF algorithm described in Section 4, (some) nodes of the SCOPE infrastructure would have been included in the VO Cometa [44]. This is due to the fact that the consumption of computational resources belonging from our bags-of-tasks, can be balanced by a corresponding amount of computational resources offered by the SCOPE infrastructure, and the inclusion of the latter into the VO Cometa could be done automatically by the GF algorithm (for this aim see Section 6.2).

6. EXPERIMENTS

In this section we present the results of a set of simulations designed to test the effectiveness of the approach. Section 6.1 contains the experimental results related to the trust measure we have introduced in Section 2. The second set of results is reported into Section 6.2 was obtained by simulating the execution of the GF algorithm discussed in Section 4, in order to analyze related performance.

DOI: 10.1002/cpe

6.1. Evaluation of the Trust model

In order to test the effectiveness of the proposed trust model we carried out some experiments involving a population consisting of 1,000 nodes which intensively interact for requiring/providing services and opinions. Each node was initially joined with a random number of Virtual Organizations comprised between 10 and 20. Each experiment consisted of 50,000 epochs, where in each epoch 250 nodes (i.e. the 25% of the overall nodes) send Grid jobs which require computational resources to other nodes of their grids. Each requiring node was able to ask an opinion to up to other 125 nodes chosen in a random way among the node population (i.e. the 12.5% of the entire population). At the end of each experiment, each node used resources of other nodes about $12 \div 13$ times in average.

As previously described into Section 2.5, a value of trust equal or greater than 0.5 identifies a trusted (i.e. honest) node and, vice versa, a trust value smaller than 0.5 is associated with an untrusted node. At the beginning of the simulatio all the nodes were assumed to be not malicious and trusted, i.e. they received an initial trust value of 0.5. Clearly, the higher the number of trusted and malicious nodes are identified during the simulation, the higher will be the accuracy of the model.

A variable number of *i*) *malicious nodes* (i.e. nodes which provide untruthful opinions about other nodes), and *ii*) *unreliable nodes* (nodes providing unreliable services for the jobs allocated into its own resources) has been considered for each experiment, as follows. In particular, we simulated three different scenarios, A, B and C, as described in Table II.

Scenario	Untrusted nodes	Malicious nodes	Behavior
A	10% or 50%	NO	Always reliable or always unreliable.
B	10% or 50%	10% or 50%	Always always reliable and honest or unreliable and malicious.
C	10% or 50%	10% or 50%	Building positive reputations on services with low relevance for cheating on those with high relevance.

Table II. Scenarios simulated for the trust system

As described in Table II, the scenario A describes a grid federation on which the 10% or 50% of the nodes assume an untrusted behavior (i.e. they are unreliable in providing services). Scenario labeled B extends scenario A, such that the nodes having an unreliable behavior, will assume also a dishonest behavior in providing recommendations to other nodes. In particular, nodes which assume a malicious behavior will provide always misleading opinions by assigning a trust value which is close to 0 to reliable nodes, and a trust value of about 1 to untrusted nodes. Finally, scenario C differs from B because malicious nodes assume a more sophisticated behavior, aimed at building a positive reputation by means of services having low relevance. They will cheat this reputation in order to have assigned services having high relevance. Furthermore, in this experiment the ratio of low to high relevant services was fixed to $1 : 4$.

To analyze experimental results, we measured the percentage of nodes which correctly recognize untrusted nodes. We denote this measure as MBE (Malicious Behavior Estimation).

Figure 3 shows the results of a set of experiments carried out by simulating three different scenarios in terms of average MBE. The curves labeled *A10* and *A50* describe the trend of the MBE over the various epochs, where *A10* refers to the case where 10% of nodes assumes an untrusted behavior, and *A50* refers to the case where 50% of nodes assumes an untrusted behavior, as reported in Table II. Moreover, curves labeled *B10* and *B50* describe the trend of MBE, when 10% and 50% of nodes have the behavior specified in the scenario B, respectively. Finally, the curves labeled *C10* and *C50* refer to the case 10% and 50% of nodes assume the behavior specified for the scenario C, respectively. Finally, we avoided to represent the results produced in absence of malicious nodes because their are practically overlapped with the curve $A10$.

DOI: 10.1002/cpe

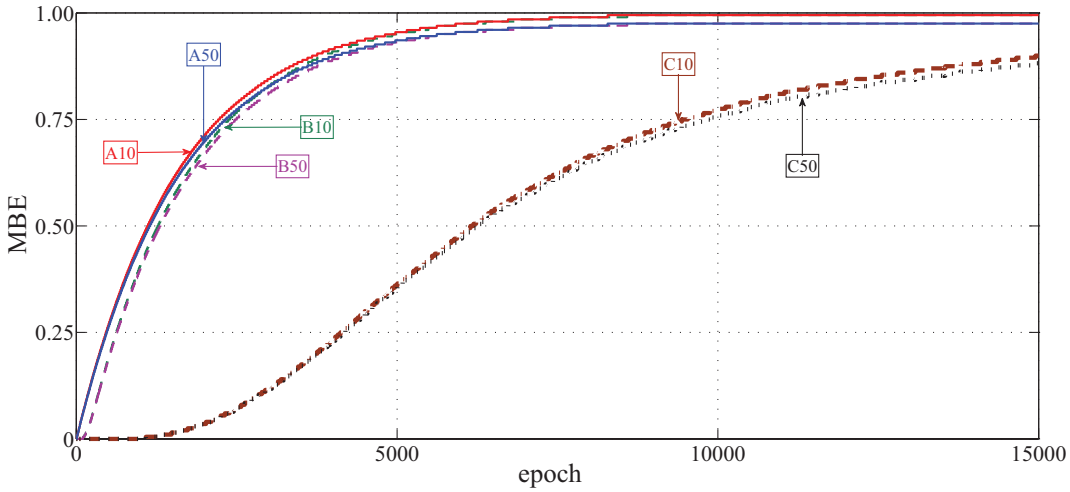

Figure 3. MBE vs epochs that for A) uniform behavior, B) alternate behavior C) misleading opinions in presence of a percentage of the 10% and 50% of malicious nodes.

By analyzing the obtained results, we can remark as each experiment highlights a significant resilience with respect to the presence of malicious and unreliable nodes. In particular, in presence of a uniform behavior (curves labeled *A10* and *A50*) the 50% (i.e. 90%) of the node population stably recognize all the malicious nodes after about 1500 (i.e. 3500) epochs. If collusive or trouble activities are introduced, the obtained results (curves labeled *B10* and *B50*) show an initial very little gap, with respect to the previous case, due to the wrong values of the malicious opinions but this gap becomes equal to 0 around the $4,500$-th epoch. Indeed, this gap is maximum when the simulation starts (see Sections 2) because *i*) the confidence factors weighting opinions are usually maximum and *ii*) the reputation has the maximum impact on the trust measures. Accordingly with the increased knowledge of each node about its environment, their confidence factors usually decrease similarly to the relevance of the reputation in computing trust measures. Finally, in presence of dynamic behaviors (curves labeled *C10* and *C50*) the influence of trouble behaviors is clearly evident. However, also in this case the gap in terms of MBE with respect the other experiments is filled up and the 90% of the nodes stably recognize all the malicious nodes after about $15,000$ epochs.

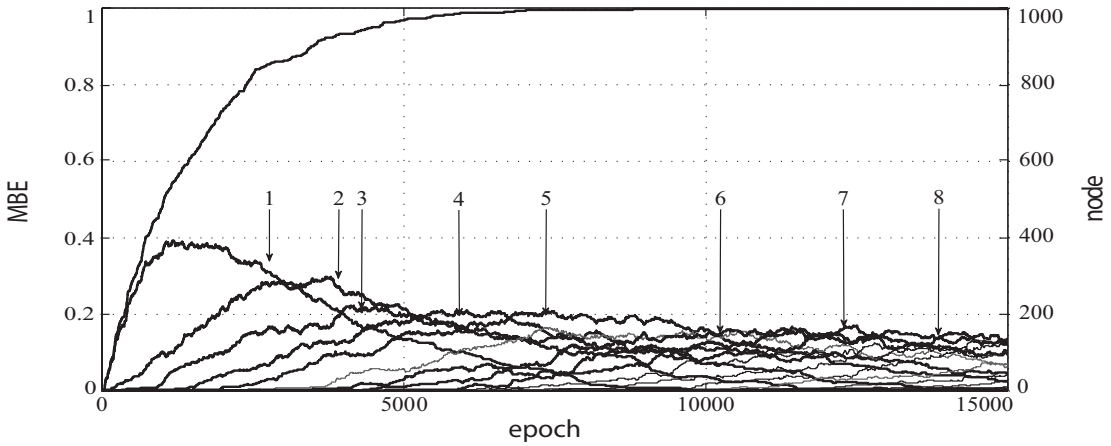

Figure 4. MBE and number of interaction carried out with the malicious node 103 vs epochs.

In figure 4 we show two measures: *i*) the percentage of the nodes which correctly recognize the behavior of the malicious node 103, i.e. the value of MBE for that particular node; *ii*) the number of

DOI: 10.1002/cpe

services assigned to the node 103 over the epochs on behalf of the considered nodes. In particular, any curve labeled x ($x = 1 \dots 8$) represents the percentage of nodes which "interacted" at most x times with the target node. From these results we can conclude that the number of interactions (i.e. submitted jobs which have been assigned to the malicious/unreliable node 103) needed to the trust model to recognize the malicious nature of the target node is very low, therefore the trust system is effective in the short term.

6.2. The GF Algorithm

The experiments described here are devoted to test the effectiveness of the proposed framework and, in particular, of the GF algorithm presented in Section 4. To this purpose we exploited another simulator, written in JAVA.

A first experiment was performed on three simulated scenarios consisting of 4000, 8000 and 16000 nodes respectively, and a fixed number of 40 grids. Moreover, 10% of nodes had an untrusted behavior. In simulating such scenarios, all the parameters were generated by means of a uniform distribution in their respective domains. Initially, we randomly generated the profile of each node which stores its orientation to consume/offer resources and its honest or malicious behavior, the number of grids which each node can be assigned (i.e. $N_{MIN} = 2$, $N_{MAX} = 10$) and the number of nodes for grid (i.e. $K_{MIN} = 10$, $K_{MAX} = 80$). These values have been chosen based on a sensitivity analysis we performed.

As a measure of the internal convenience of a grid g_j, we introduced the concept of *Average Convenience* (AC) computed on the basis of the convenience measure described by Equation 12 in Section 2. More in detail, the Average Convenience AC_j is the average of all the measures of convenience $\eta_{j,i}$ computed by the grid $g_j \in \mathcal{G}$ for all its affiliated nodes $n_i \in g_j$. Therefore, to measure the global convenience of all the grids belonging to $\mathcal{G}$ in our simulated scenario, we computed the mean (MAC) and the standard deviation (DAC) of all the AC_j:

$$MAC = \frac{\sum_{g_j \in G} AC_j}{||G||} \qquad DAC = \sqrt{\frac{\sum_{g_j \in G} (AC_j - MAC)^2}{||G||}}$$

Average convenience. The contribution of the GF algorithm in improving the performance of the Grid Federation increases when the convenience measures tend to become stable and, as previously specified, this mainly depends on the trust model. Therefore, in order to measure the advantages introduced by the GF algorithm, we set the simulator to start the execution of the GF algorithm at the 100-th epoch, when the level of stability of the convenience measures is still low due to a limited number of performed interactions (each node provided about 15 services[§], and executed for 20 epochs, because, as we will show below, MAC and DAC values reach stable values very quickly.

The initial values for the MAC and DAC measures when the GF algorithm starts to run are summarized into Tables III-a and III-b, along with the relevant parameters.

Sc.	Nodes	Grids	MAC	DAC
1	4000	40	0.156	0.0032
2	8000	40	0.156	0.0033
3	16000	40	0.155	0.0035

GF starts exec. at	K_{min}	K_{Max}	N_{min}	N_{Max}
100 epoch	10	80	2	10

Table III. MAC, DAC and parameters for the 1st experiment (effect of the GF algorithm on the convenience)

The results of the simulations for each of the three scenarios, in terms of MAC (DAC) with respect to the epochs, are shown in the left (right) part of Figures 5-7.

[§]Here a service can assume the meaning of a bag of jobs/tasks submitted to the local resource manager by the resource broker

DOI: 10.1002/cpe

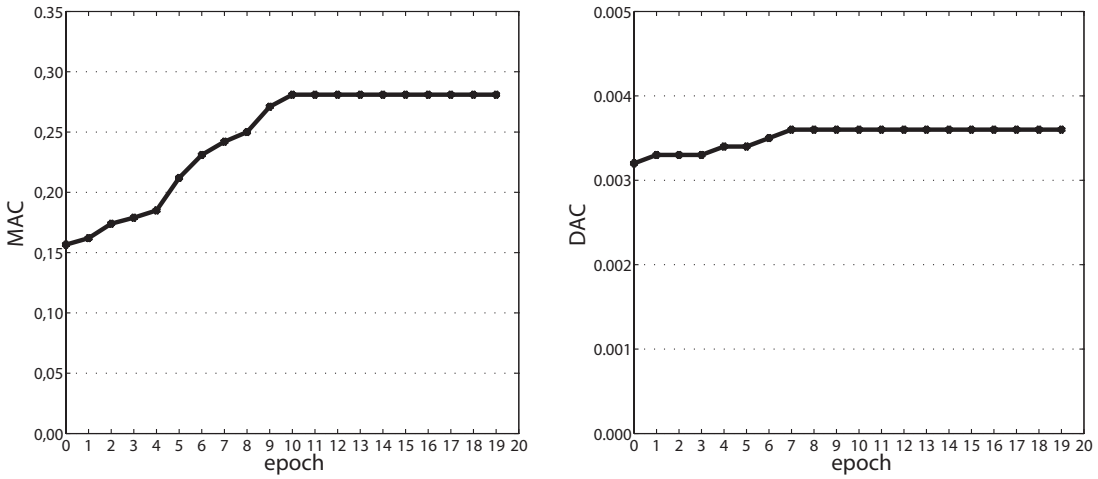

Figure 5. Experiment 1, 40 grids: Variation of MAC (left) and DAC (right) vs epochs in the 1st scenario, with 4000 nodes.

The results clearly point out that the GF algorithm introduces a significant increment of the convenience of the grids, that after a period of only 10 epochs achieves a stable configuration, where $MAC = 0.281$ and $DAC = 0.0035$ in the first scenario, $MAC = 0.295$ and $DAC = 0.0036$ in the second scenario and $MAC = 0.305$ and $DAC = 0.0041$ in the third scenario. Therefore, we have obtained an improvement of about a 44% (resp. 47, 49%) in average convenience of the grids in the first (resp. second, third) scenario, while the standard deviation from this compactness value remains small enough. We also observe that the performances of the algorithm improves when the number of nodes increases too.

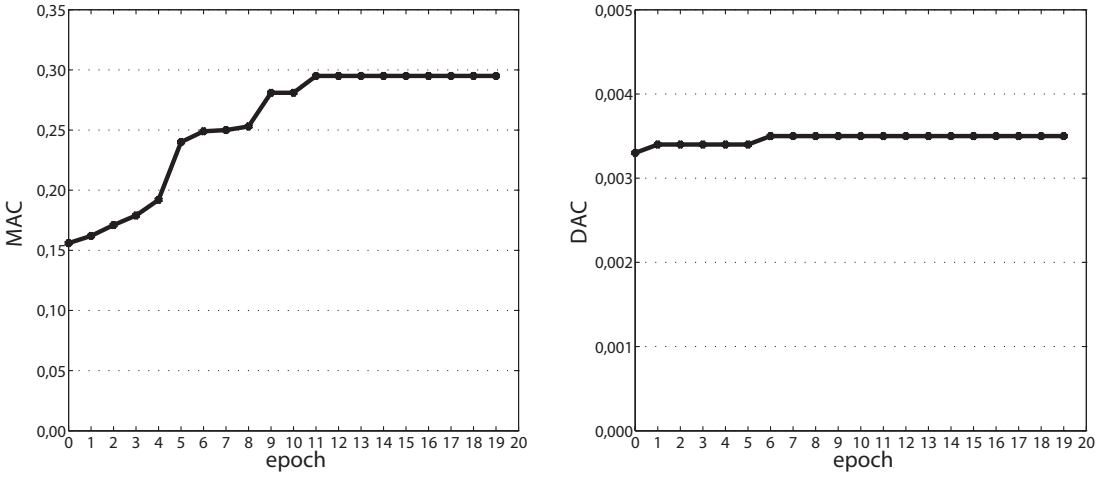

Figure 6. Experiment 1, 40 grids: Variation of MAC (left) and DAC (right) vs epochs in the 2nd scenario, with 8000 nodes.

Average convenience vs no. of Grids. In a second experiment, in order to investigate the effect of an increasing number of grids on the algorithm effectiveness, we considered other three simulated scenarios having a fixed number of 16.000 nodes, and three different numbers of grids, namely 80, 160 and 320, respectively. Also in this case we have simulated a random configurations for the grid.

DOI: 10.1002/cpe

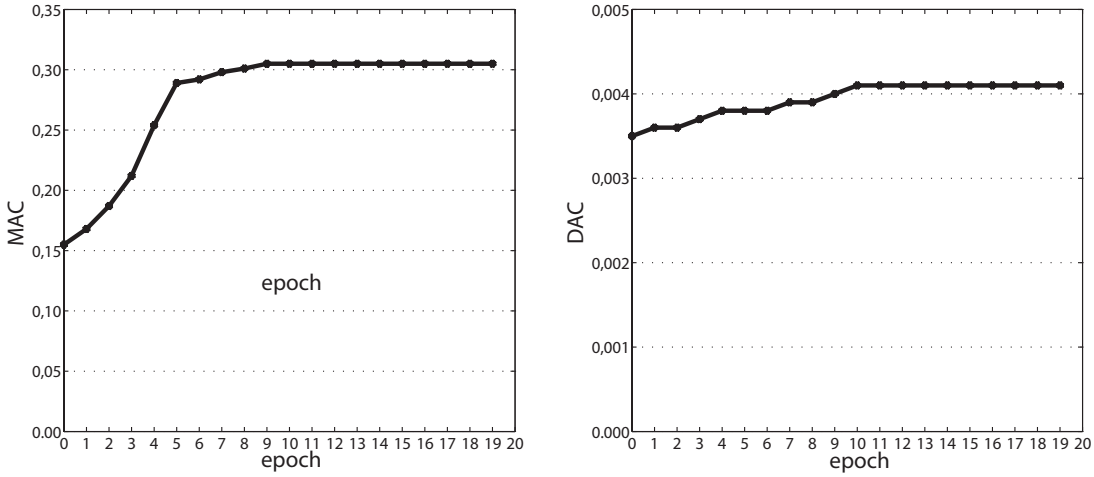

Figure 7. Experiment 1, 40 grids: Variation of MAC (left) and DAC (right) vs epochs in the 3rd scenario, with 16000 nodes.

The initial values for the compactness measures are shown into tables IV, along with the remaining parameters (shown in the table in the right), which remained unchanged.

Sc.	Nodes	Grids	MAC	DAC
1	16000	80	0.158	0.0037
2	16000	160	0.161	0.0039
3	16000	320	0.165	0.0040

GF starts exec. at	K_{min}	K_{Max}	N_{min}	N_{Max}
100 epoch	10	80	2	10

Table IV. MAC, DAC and parameters for the 2nd experiment (increasing number of Grids vs convenience)

The results of the simulations for each of the three scenarios, in terms of MAC (DAC) with respect to the epochs, are shown in the left (right) part of Figures 8-10.

We remark that the GF algorithm improves its performances in term of convenience when the number of grids increases. In particular, after a period of 10 epochs, it achieves a stable configuration where $MAC = 0.311$ and $DAC = 0.0043$ in the first scenario, $MAC = 0.327$ and $DAC = 0.0046$ in the second scenario and $MAC = 0.350$ and $DAC = 0.0052$ in the third scenario. As a result, we obtained an improvement of about a 49% (resp. 51, 43%) in average convenience of the grids in the first (resp. second, third) scenario, while also in this case the standard deviation from this convenience value remains small.

7. RELATED WORK

The Grid Federation paradigm extends the classical grid architecture by connecting computational nodes of different grids. Coupling a great number of computing nodes at large scale enforces to engage several challenges for coordinating different grid policies, e.g. resource discovery, task scheduling, QoS, fault tolerance and security issues [46]. Therefore, the approach of constituting federated grids leads to develop a variety of different approaches to perform, e.g., inter-grid scheduling, resource allocation and coordination of grid brokers. In the following we discuss the approaches which, to the best of our knowledge, gave a contribution in the area we have addressed in our work, i.e. resource allocation and trust systems in grid federation, by discussing the original aspects of our approach.

DOI: 10.1002/cpe

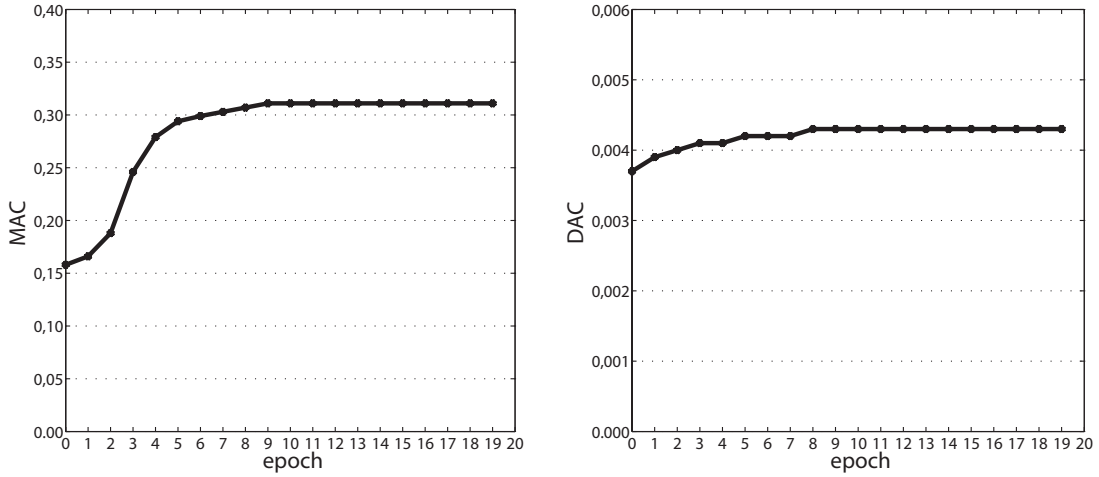

Figure 8. Experiment 2, 16000 nodes: Variation of MAC (left) and DAC (right) vs epochs in the 1st scenario, with 4000 nodes.

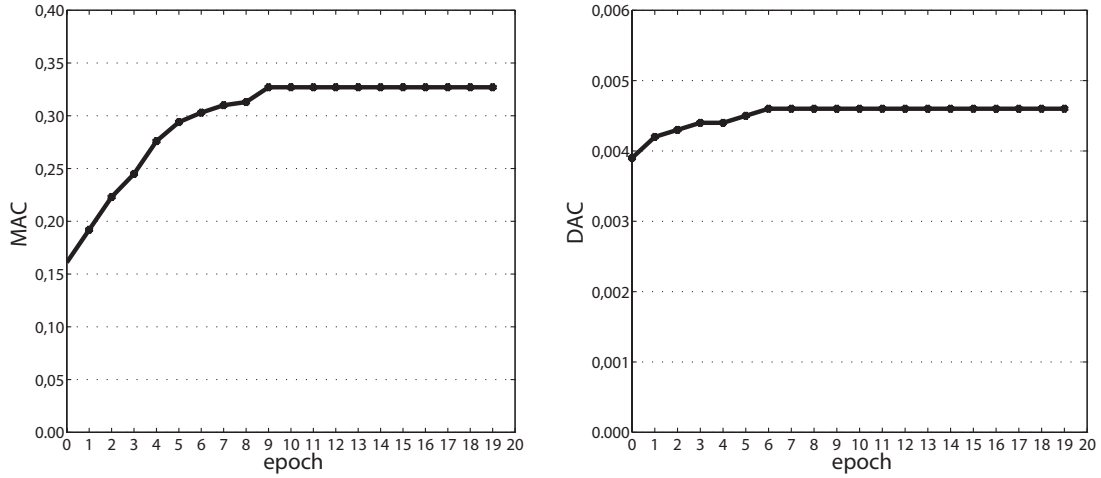

Figure 9. Experiment 2, 16000 nodes: Variation of MAC (left) and DAC (right) vs epochs in the 2nd scenario, with 8000 nodes.

Finally, at the end of this Section we briefly discuss how this work is partially based on the preliminary study presented in [23].

Task Scheduling and resource allocation in Grid Federations. Economical approaches [47] have been used in the past to allocate computational resources in distributed environments in a scalable and adaptable way [48, 49]. In particular, auctions mechanisms, which are essentially based on the equilibrium between resource demands and supplies, have shown their capability in efficiently allocating resources. Prices are established by the market and take into account QoS requirements [50]. Similarly, authors of [14] proposed to model a Grid Federation as an economy driven large scale scheduling system. Indeed, they designed an algorithm able to improve the efficient allocation of jobs on cluster resources across the grid federation and to satisfy the QoS constraints prescribed by the resource consumers. An analogous approach is adopted in [51] where a "computational economy methodology" for a federation of grid agents is discussed. More in detail, a QoS scheduling of distributed clusters of resources in a cooperative fashion is implemented.

Prepared using cpeauth.cls
DOI: 10.1002/cpe

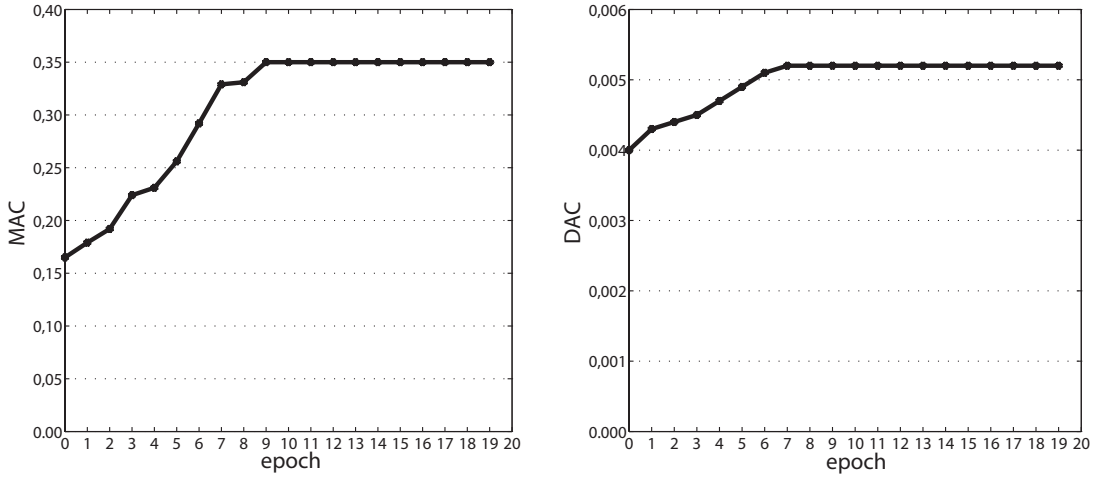

Figure 10. Experiment 2, 16000 nodes: Variation of MAC (left) and DAC (right) vs epochs in the 3rd scenario, with 16000 nodes.

DRIVE [52] is another proposal for a distributed economic meta-scheduler for a trustworthy allocation of the cost of resources across the members of a Virtual Organization. A major trait of DRIVE is its ability to support a wide range of topologies from small scale local grids through global Grid Federations. Resources are allocated by outlining the requirements of each service and then negotiated by using a two stage contract. In the first stage, the service is allocated and negotiated but the resources availability is not guarantee, while the final contract stage is performed by a further SLA negotiation which fixes all the terms of the services. In this way, the allocation latency is reduced and the negotiation participation is supported.

In [53] Grid Federation is devoted to construct an utility computing infrastructure. The work is based on the adoption of the Globus Toolkit [54] and consolidated standards in order to provide a full meta-scheduling system which shows flexibility and scalability properties. The authors state that the proposed solution exhibits many advantages (security, scalability, etc.) and good performances, particularly in presence of intensive computing applications. In [55] the performances of cooperative, semi-cooperative and non-cooperative grid resource allocation mechanisms, based on the game theory, are compared and analyzed by exploiting an extended set of experiments. The adopted simulation model has been realized around large grid sites connected together as a Federation, which accepts job to be scheduled by a main portal giving the access to the Federation.

Authors of [56] focused on scheduling strategies for HTC (High Throughput Computing) applications applied to federated grids. They classify infrastructures forming a federated grid as internal or external, in order to apply a scheduling policy which allocates resources internally, i.e. into its own grid infrastructure. On one hand, their approach is devoted to save time and communication bandwidth by exploiting the location of the internal resources and the membership to their respective resource domains. They present three novel algorithms for inter-grid scheduling in presence of enterprise, partner, or utility grid infrastructures. Policies presented in their approaches are interesting because they are aimed to limit overload of external resources, such resource sharing between different grid organizations is facilitated.

Interlinking of grids through peering arrangements, aimed at enabling inter-grid resource sharing, is proposed in [57]. The authors focus on the fact that the resulting large scale infrastructure is potentially able to grow in a sustainable way. For this aim, they also propose a reference architecture, including inter-grid gateways, and a set of mechanisms, some policies to allow the inter-networking of grids. A further, remarkable example in the area of interlinked grid is reported in [58]. In that paper, the author consider an interlinked cloud model in which multiple clouds are available and collaborate. Cloud platforms may differ both in performance and size and each

DOI: 10.1002/cpe

platform provides its own job scheduling criteria. Two algorithms (both derived from Simulated Annealing) are considered to perform job scheduling. In the above contributions, the problem of optimizing the QoS provided by the grid nodes is central. [59]. HDCF platforms significantly differ from other grid architectures in terms of the strategy they adopt to establish federation and pricing policies. In HCDFs, different providers collaborate on the basis of specific demands to achieve economies of scale and to best fit client requirements. The authors of [59] suggest to apply tools from Cooperative Game Theory to allocate computational resources. In detail, they focused on the problem of motivating a provider (which is modelled as a self-interested entity) to form or join an existing HCDF and attempted at measuring the amount of resources required to fulfill such a goal. The authors of [59] proposed two utility maximizing cooperative games: the former allowed for maximizing the total profit while the latter was instrumental in maximizing social welfare.

A nice approach to job scheduling in a Grid federation is discussed in [60]. In that approach, the authors deal with a computational environment mainly formed by storage units (which are in charge of receiving and storing the data collected by remote sensing ground station) and computational resources (which are in charge of processing input data). Such an architecture provides also a data center which is able to provide both computational and storage capabilities. Once some tasks are submitted to a resource broker, a scheduler is invoked to define a suitable workflow as well as to assign tasks to resources. Tasks and their dependencies are represented by means of an *hypergraph* $\mathcal{H}$ such each task t_i is represented by a vertex v_i and each hyperedge e_j represents a file f_j required by tasks. The problem of allocating tasks to computational resources is then formulated as the problem of partitioning G into clusters under the assumption that the number K of groups is less than the number of computational resources. The partitioning task has to be performed with the goal of evenly distributing load among groups.

Most of the existing approaches to job scheduling in grids consider only the incentives to favor a specific part (e.g., users or resource providers). Very few studies deal with job scheduling by optimizing the incentives to both users and providers. To this end we cite a very recent work [61], in which the authors describe a multi-objective optimization approach seeking at jointly maximizing the successful execution rate of jobs and minimizing the incentives for grid users and resource providers). To solve this optimization problem, a greedy scheduling algorithm is presented.

Summarizing the scope of those approaches, it can be stated that they take into account the result of an utility function, coming or not from an explicit economical model, which leads to assign a value (let be real or virtual) to a strategy or an algorithm to schedule tasks and/or assign resources to grid users, i.e. activities devoted to manage and control the Grid Federation. On the other hand, the main novelty of our approach is that VOs identities and goals are preserved, such that the federated grids are fully compatible with the original goals of the federated Virtual Organizations. Indeed, the presented approach includes mechanisms aimed at improving the global utility offered by each grid to its users, which constitute the primary goal of every Virtual Organizations [1]. This is performed by defining the problem of Grid Formation in the dynamic context of Open Grid Federation. The proposed solution is defined around the concept of "convenience" (convenience measure).

Trust systems for Grid VOs and Grid Federations. In proposing our solution to the grid formation problem, we rely on a distributed trust system which, in fact, constitutes a key contribution in the computation of the convenience measure.

Prior to discussing related literature on trust management and grid system, we wish to highlight that trust is now an interdisciplinary concept. Relevant scientific advancements have been achieved in the management of trust among humans (e.g., in case of Online Communities) and some of these results can be exported to the field of Grid Management and Formation. For instance, we recently studied how to dynamically form groups in Online Social Networks [62]. We studied such a problem from two different but related perspectives, i.e., we assessed the convenience of a user in joining with a group and we targeted at checking the advantages a group can get when the group administrator decides to admit a new member. The problem of matching users with groups (and vice versa) resembles the problem of deciding which grids a computational node may join and in both cases, we need to optimize a suitable objective function. However, the coefficients of each objective function as well as associated constraints strictly depend on the application scenario: in case of a

DOI: 10.1002/cpe

grid, in fact, we should pay attention to the hardware/software features of involved nodes while in case of humans we must record their past behaviors to decide the suitability of a user for a given group. More in general, the analysis of past user behavior to satisfy her/his information needs as well as to provide recommendation is now a mature research area [63].

In this field some past approaches aimed at managing resources and task allocation by exploiting reliability and reputation information, often synthesized in a single unique trust measure [64, 65, 66, 67, 68].

Furthermore, in order to deal with unreliability of computing resources at large scale, decentralized grid overlays are proposed together with a reputation-based grid workflow scheduling algorithm in [69]. The proposed algorithm allows the grid federation to dynamically adapt itself to changes occurring in resource conditions and in dealing with unsuccessful job execution events or resource failures. The scheduling algorithm is based on global and local information about reputation, which are considered statistical properties, and are obtained by exploiting feedbacks automatically computed on the basis of the results of each performed transaction.

The opportunities provided by trust systems are exploited also in [13], on which a novel framework, called Trust-Incentive resource Management, is designed to take into account the management of grid resources by introducing values of prices, trust incentives, and a weighted voting scheme. Providers set the price by according to demand and supply, and consumers maximize the surplus upon budget and deadline, while the weighted voting scheme secures the grid by declining join requests coming from malicious nodes. In this approach the trust model is very simple, as the trust score is obtained by considering only the number of direct and indirect experiences. Authors present also a set of experimental results which confirm improvements in terms of resource allocation efficiency, system completion time, and aggregated resource utilization.

The reputation management system presented in [70] – called "Network of Favors" – is interleaved in a grid context in order to implement a one-to-one resource sharing credit between the resource providers. In particular, a node A which receives from B a request to use its own resources, will calculate the reputation of a node B by using the values of "favors" A has received from B and vice versa. Similarly to [69] only the overall success or fail in providing a service is considered. Authors addressed different strategies against to malicious behaviors: *i*) identity changes are contrasted by considering in the reputation also the past interactions history by differentiating long-known nodes by newcomers and by using cryptography techniques; *ii*) the work validity is assured by avoiding reply strategies; *iii*) denial-of-service attacks are warded off by using sandboxes with restricted access to the underlying machine, and no network accesses. Nevertheless, such an approach is similar to that adopted by eBay, therefore it suffers of the same vulnerabilities [71].

In all the approaches discussed above, trust and reputation are exploited in the context of computational grids in order to allow a better management and sharing of computational resources (e.g. by assigning resources and services and/or monitoring specific behaviors). Such approaches implement automatic mechanisms to introduce feedbacks in front of services, and simple strategies which only consider the number of transactions carried out successfully. The trust model we designed for our proposal allows software agents to evaluate relevance and feedback of provided service. In addition, it allows to consider all the aspect involved in the service provisioning, such that the result is a fine-grained evaluation of the resulting QoS.

Similarly, when direct and indirect reliability measures are adopted to construct the trust score, any of these approaches modulate their reciprocal relevances on the basis of the knowledge hold by a node about another one. Moreover, in computing reputation measures only our proposal provides a different weight to the opinions of the less affordable nodes. Differently by our proposal, in order to contrast malicious activities we may refer to the proposal in [69], which implements some suitable strategies in the trust system. As shown in the experimental section, our trust system is able to effectively contrast the effects due to an high presence of malicious nodes in the system and quickly allows their identification.

As stated in Section 1, the approach described in this work is partially based on a preliminary study presented in [23]. In particular, although the definitions of the behavioral measures are similar, that presented in this work is generic, i.e. it takes into account of all the possible cost components.

DOI: 10.1002/cpe

The trust model defined in [23], which is a significant part of the convenience measure, was only a short and preliminary version of that one presented in this work. Indeed, it considered only the feedbacks provided by the grid users, while in this study we take into account two distinct measures, reliability and reputation, which are carefully designed and discussed. In the present work, the trust model includes several measures in order to take into account the freshness of trust information, and to limit malicious behaviors of software agents. In the experimental Section of this work an extensive set of results is presented and discussed to validate our approach. The extensive discussion about related work – contents of this section – is original and based on the model and experimental results presented in the previous Sections.

8. CONCLUSIONS

In this work we proposed a model to improve the QoS provided by grid nodes in an open and dynamic federation of Grid VOs. We defined the reference context as *Open Grid Federation* and we dealt with the problem of Grid Formation, i.e. which nodes should be assigned to grids with the goal of optimizing the overall QoS provided within its own grid. Our solution exploits software agents which compute behavioral and trust measures on behalf of grids and nodes. The GF algorithm, designed at this purpose, is aimed at balancing demand and offer of resources in order to avoid unsatisfied requests and unallocated resources. It makes use of an integrated measure, i.e. the convenience measure, resulting from two distinct measures: *i*) the nodes (past) behaviors, in terms of costs of the resources requested/offered and *ii*) the trustworthiness of nodes, which is based on data automatically collected by software agents assisting grid nodes. In particular, trust measures we take into account in our work include a fine grained evaluation of the relevance of the services mutually provided within each grid VO. The result is an ad hoc trust model which is suitable to be combined with behavioral measures to quickly obtain reliable convenience measures. At the best of our knowledge, the presented approach is the first framework which uses a trust system to manage the affiliation of a node with a grid.

In order to measure the effects exploited by the application of our proposal, we performed extensive simulations. As shown in the experimental results, the algorithm is able to quickly and dynamically assigns nodes to grids and vice versa, even in presence of large grids, by solving a suitable matching problem in a distributed fashion.

ACKNOWLEDGEMENTS

This work has been partially supported by the following projects

- **PRISMA** PON04a2 A/F funded by the Italian Ministry of Education, University, and Research
- **TENACE PRIN Project** (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research,
- Program "Programma Operativo Nazionale Ricerca e Competitività" 2007-2013, project **BA2Kno** (Business Analytics to Know) PON03PE_00001_1, in "Laboratorio in Rete di Service Innovation",
- Program "Programma Operativo Nazionale Ricerca e Competitività" 2007-2013, **Distretto Tecnologico CyberSecurity** funded by the Italian Ministry of Education, University and Research.

REFERENCES

1. Foster I, Kesselman C. *The Grid 2: Blueprint for a new computing infrastructure*. Elsevier, 2003.
2. EMI. European Middleware Initiative. http://www.eu-emi.eu.
3. Hoschek W, Jaen-Martinez J, Samar A, Stockinger H, Stockinger K. Data management in an international data grid project. *Grid ComputingGRID 2000*. Springer, 2000; 77–90.
4. Lamanna M. The LHC computing grid project at CERN. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 2004; **534**(1):1–6.

DOI: 10.1002/cpe

5. Ellert M, Konstantinov A, Kónya B, Smirnova O, Wäänänen A. The nordugrid project: Using globus toolkit for building grid infrastructure. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 2003; **502**(2):407–410.
6. Buyya R, Ranjan R. Special section: Federated resource management in grid and cloud computing systems. *Future Generation Computer Systems* 2010; **26**(8):1189–1191.
7. Jie W, Arshad J, Sinnott R, Townend P, Lei Z. A review of grid authentication and authorization technologies and support for federated access control. *ACM Computing Surveys (CSUR)* 2011; **43**(2):12.
8. Leal K. Self-adjusting resource sharing policies in federated grids. *Future Generation Computer Systems* 2013; **29**(2):488–496.
9. Salehi MA, Buyya R. Contention-aware resource management system in a virtualized grid federation. student research symposium of 17th IEEE International Conference on High Performance Computing (HiPC 11), India, 2011.
10. Ranjan R, Harwood A, Buyya R. Coordinated load management in peer-to-peer coupled federated grid systems. *The Journal of Supercomputing* 2012; **61**(2):292–316.
11. Williams D, Bell G, Cinquini L, Fox P, Harney J, Goldstone R. Earth system grid federation: Federated and integrated climate data from multiple sources. *Earth System Modelling-Volume 6*. Springer, 2013; 61–77.
12. Suhardiman B. Support e-science activities in indonesia. *Proceedings of The International Symposium on Grids and Clouds (ICGC 2012). 26 February-2 March. Taipei, Taiwan*, vol. 1, 2012; 44.
13. Zhang Y, Huai J, Liu Y, Lin L, Yang B. A framework to provide trust and incentive in crown grid for dynamic resource management. *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings. 15th International Conference on*, IEEE, 2006; 390–395.
14. Ranjan R, Harwood A, Buyya R, *et al.*. Grid federation: An economy based, scalable distributed resource management system for large-scale resource coupling. *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia* 2004; .
15. Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organizations. *International journal of high performance computing applications* 2001; **15**(3):200–222.
16. Kivity A, Kamay Y, Laor D, Lublin U, Liguori A. KVM: the linux virtual machine monitor. *Proc. of the Linux Symp.*, vol. 1, 2007; 225–230.
17. 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* 2003; **37**(5):164–177.
18. Sefraoui O, Aissaoui M, Eleuldj M. Openstack: toward an open-source solution for cloud computing. *International Journal of Computer Applications* 2012; **55**(3):38–42.
19. Comi A, Fotia L, Messina F, Rosaci D, Sarnè GM. A qos-aware, trust-based aggregation model for grid federations. *On the Move to Meaningful Internet Systems: OTM 2014 Conferences (CooPIS - Cooperative Information Systems)*. Springer Berlin Heidelberg, 2014; 277–294.
20. Messina F, Pappalardo G, Tramontana E. Design and evaluation of a high-level grid communication infrastructure. *Concurrency and Computation: Practice and Experience* 2007 DOI 101002/cpe1104; **19**(9):1299–1316.
21. Giunta R, Messina F, Pappalardo G, Tramontana E. Providing qos strategies and cloud-integration to web servers by means of aspects. *Concurrency and Computation: Practice and Experience* 2013 DOI 101002/cpe3031; **27**(6):1498–1512.
22. Sabater J, Sierra C. REGRET: Reputation in Gregarious Societies. *AGENTS '01: Proc. of the Fifth Int. Conf. on Autonomous Agents*, ACM Press: New York, NY, USA., 2001; 194–195.
23. De Meo P, Messina F, Rosaci D, Sarné G M L. Improving grid nodes coalitions by using reputation. *Intelligent Distributed Computing VIII*, *Studies in Computational Intelligence*, vol. 570, Camacho D, Braubach L, Venticinque S, Badica C (eds.). Springer International Publishing, 2015; 137–146.
24. IGI. Italian Grid Infrastructure. www.italiangrid.it.
25. Laure E, Edlund A, Pacini F, Buncic P, Barroso M, Di Meglio A, Prelz F, Frohner A, Mulmo O, Krenek A, *et al.*. Programming the grid with gLite. *Technical Report* 2006.
26. Krauter K, Buyya R, Maheswaran M. A taxonomy and survey of grid resource management systems for distributed computing. *Software: Practice and Experience* 2002; **32**(2):135–164.
27. Torque resource manager. http://www.adaptivecomputing.com/.
28. Messina F, Pappalardo G, Santoro C. Complexsim: a flexible simulation platform for complex systems. *International Journal of Simulation and Process Modelling* 2013 DOI 101504/IJSPM2013059417; **8**(4):202–211.
29. Yu J, Buyya R. A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing* 2005; **3**(3-4):171–200.
30. Netto MA, Buyya R. Coordinated rescheduling of bag-of-tasks for executions on multiple resource providers. *Concurrency and Computation: Practice and Experience* 2012; **24**(12):1362–1376.
31. Foster I, Kesselman C, Nick JM, Tuecke S. Grid services for distributed system integration. *Computer* 2002; **35**(6):37–46.
32. Rosenblum M. Vmwares virtual platform. *Proceedings of hot chips*, vol. 1999, 1999; 185–196.
33. Wooldridge M, Jennings NR. Intelligent agents: Theory and practice. *The knowledge engineering review* 1995; **10**(02):115–152.
34. Grandison T, Sloman M. Trust management tools for internet applications. *Trust Management*. Springer, 2003; 91–107.
35. Abdul-Rahman A, Hailes S. Supporting trust in virtual communities. *HICSS '00: Proc. of the 33rd Hawaii Int. Conf. on System Sciences-Volume 6.*, vol. 6, IEEE Computer Society.: Washington, DC, USA, 2000.
36. Netto MA, Buyya R. Coordinated rescheduling of bag-of-tasks for executions on multiple resource providers. *Concurrency and Computation: Practice and Experience* 2012; **24**(12):1362–1376, doi:10.1002/cpe.1841.
37. Cirne W, Brasileiro F, Sauvé J, Andrade N, Paranhos D, Santos-neto E, Medeiros R, Gr FC. Grid computing for bag of tasks applications. *In Proc. of the 3rd IFIP Conference on E-Commerce, E-Business and EGovernment*, Citeseer, 2003.

38. Ramchurn S, Huynh D, Jennings N. Trust in multi-agent systems. *Knowledge Engeenering Review* 2004; **19**(1):1–25.
39. Zacharia G, Maes P. Trust management through reputation mechanisms. *Applied Artificial Intelligence.* 2000; **14**(9):881–907.
40. Messina F, Pappalardo G, Santoro C. A self-organising system for resource finding in large-scale computational grids. *WOA 2010 - XI Workshop Nazionale "Dagli Oggetti agli agenti"*, 2010.
41. Messina F, Pappalardo G, Santoro C. Hygra: A decentralized protocol for resource discovery and job allocation in large computational grids. *Computers and Communications (ISCC), 2010 IEEE Symposium on*, IEEE, 2010; 817–823.
42. Messina F, Pappalardo G, Santoro C. Exploiting the small-world effect for resource finding in p2p grids/clouds. *20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE, 2011 DOI 10.1109/WETICE.2011.22; 122–127.
43. Messina F, Pappalardo G, Santoro C. Decentralised resource finding in cloud/grid computing environments: A performance evaluation. *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, IEEE, 2012 DOI 10.1109/WETICE.2012.111; 143–148.
44. Consorzio Cometa. http://www.consorzio-cometa.it/.
45. The SCOPE project and DATACENTER. http://www.scope.unina.it/.
46. Antoniadis P, Fdida S, Friedman T, Misra V. Federation of virtualized infrastructures: sharing the value of diversity. *Proceedings of the 6th International COnference*, ACM, 2010; 12.
47. Sutherlaind I. A futures market in computer time 1968; .
48. Buyya R, Abramson D, Giddy J, Stockinger H. Economic models for resource management and scheduling in grid computing. *Concurrency and computation: practice and experience* 2002; **14**(13-15):1507–1542.
49. Chien CH, Chang P, Soo VW. Market-oriented multiple resource scheduling in grid computing environments. *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, vol. 1, IEEE, 2005; 867–872.
50. Krawczyk S, Bubendorfer K. Grid resource allocation: allocation mechanisms and utilisation patterns. *Proceedings of the sixth Australasian workshop on Grid computing and e-research-Volume 82*, Australian Computer Society, Inc., 2008; 73–81.
51. Ranjan R, Buyya R, Harwood A. A case for cooperative and incentive-based coupling of distributed clusters. *Cluster Computing, 2005. IEEE International*, 2005; 1–11.
52. Chard K. Drive: A distributed economic meta-scheduler for the federation of grid and cloud systems 2011; .
53. Vázquez T, Huedo E, Montero R, Llorente I. Evaluation of a utility computing model based on the federation of grid infrastructures. *Euro-Par 2007 Parallel Processing*. Springer, 2007; 372–381.
54. The Globus Alliance. The globus toolkit 2014. http://toolkit.globus.org/toolkit/.
55. Khan S, Ahmad I. Non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation. *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, IEEE, 2006; 10–pp.
56. Leal K, Huedo E, Llorente IM. Performance-based scheduling strategies for htc applications in complex federated grids. *Concurrency and Computation: Practice and Experience* 2010; **22**(11):1416–1432.
57. Dias de Assuno M, Buyya R, Venugopal S. Intergrid: a case for internetworking islands of grids. *Concurrency and Computation: Practice and Experience* 2008; **20**(8):997–1024, doi:10.1002/cpe.1249.
58. Moschakis I, Karatza H. Multi-criteria scheduling of bag-of-tasks applications on heterogeneous interlinked clouds with simulated annealing. *Journal of Systems and Software* 2015; **101**:1–14.
59. Hassan M, Hossain M, Sarkar A, Huh E. Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform. *Information Systems Frontiers* 2014; **16**(4):523–542.
60. Zhang W, Wang L, Ma Y, Liu D. Design and implementation of task scheduling strategies for massive remote sensing data processing across multiple data centers. *Software: Practice and Experience* 2014; **44**(7):873–886.
61. Xu H, Yang B. An incentive-based heuristic job scheduling algorithm for utility grids. *Future Generation Computer Systems* 2015; **49**:1–7.
62. De Meo P, Ferrara E, Rosaci D, Sarnè GML. Trust and compactness in social network groups. *IEEE Transactions on Cybernetics* 2015; **45**(2):205–216.
63. De Meo P, Quattrone G, Ursino D. Integration of the HL7 standard in a multiagent system to support personalized access to e-health services. *IEEE Transactions on Knowledge and Data Engineering* 2011; **23**(8):1244–1260.
64. Aberer K, Despetovic Z. Managing trust in peer-2-peer information systems. *Proc. of Information and Knowledge Management, 10th International Conference on*, ACM, 2001; 310–317.
65. Huynh T, Jennings N, Shadbolt N. An integrated trust and reputation model for open multi-agent system. *Autonmous Agent and Multi Agent Systems* 2006; **13**:119–154.
66. Rosaci D, Sarnè G M L, Garruzzo S. Integrating trust measures in multiagent systems. *International Journal of Intelligent Systems* 2012; **27**(1):1–15.
67. Xiong L, Liu L. Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transaction on Knowledge and Data Engineering* 2004; **16**(7):843–857.
68. Birk A. Boosting Cooperation by Evolving Trust. *Applied Artificial Intelligence.* 2000; **14**(8):769–784.
69. Rahman M, Ranjan R, Buyya R. Dependable workflow scheduling in global grids. *Grid Computing, 2009 10th IEEE/ACM International Conference on*, IEEE, 2009; 153–162.
70. Andrade N, Brasileiro F, Cirne W, Mowbray M. Discouraging free riding in a peer-to-peer cpu-sharing grid. *High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on*, IEEE, 2004; 129–137.
71. Resnick P, Zeckhauser R. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in applied microeconomics* 2002; **11**:127–157.