Reputation-Based Grid Resource Selection

Beulah Kurian Alunkal,^{1,2} Ivana Veljkovic,^{1,3}
Gregor von Laszewski,^{1,*} and Kaizar Amin^{1,4}

[1] Argonne National Laboratory, Argonne, IL 60439, U.S.A.
[2] Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, U.S.A.
[3] Department of Computer Science and Engineering, The Pennsylvania State University, PA 16802, U.S.A.
[4] Department of Computer Science and Engineering, University of North Texas, TX 76201, U.S.A.
[*] Corresponding Author: gregor@mcs.anl.gov

***Abstract*— The Grid approach provides the ability to access, utilize, and manage a variety of heterogeneous resources in virtual organizations across multiple domains and institutions. Selecting appropriate resources within such a distributed Grid environment to satisfy quality of service requirements is a complex and difficult task. This paper proposes a reputation management framework for Grids to facilitate a distributed and efficient mechanism for resource selection. Our reputation management service is based on the concept of dynamic trust and reputation adaptation based on community experiences to classify, select, and tune the allocation of entities, including resources, service, and services provided by people. The framework can evaluate through specialized services simple contextual quality statements in order to effect the reputation for a monitored resource. The proposed reputation service uses a novel algorithm for evaluating Grid reputation by combining two known concepts (a) using eigenvectors to compute reputation and (b) integrating global trust. We the resulting framework *GridEigenTrust framework*.**

I. Introduction

Grid computing [1] initially focused on large-scale resource sharing, innovative applications, and achievement of high-performance. Today, the Grid approach [2] suggests the development of a distributed service environment that integrates a wide variety of resources with various quality of service capabilities to support scientific and business problem solving environments. However, optimal utilization of these distributed services and resources often require the Grid user to make a prudent decision regarding the capacity of these remote resources. Users are faced with questions such as: which resources are available remotely, what capabilities do these resources have, am I authorized to use these resources, and on which resources do I have the chance to execute my tasks with the most success?

In a typical Grid scenario users are interested in identifying possible candidate resources through meta information that is obtained from directories, databases, or registries. However, the current generation of Grid information services provides only the most elementary information to guide a more sophisticated quality of service based resource selection process. The Globus Monitoring and Directory Service (MDS) [3] provides a limited set of information about Grid resources including static and possibly dynamic properties. In many cases the information returned by this service is costly to obtain, inaccurate or outdated, and does not integrate a resource selection service. Additionally, we often lack information in regards to a metric that provides information about the quality of the provided entities similar to an Internet shopping site, which classifies included items while augmenting them with information in regards to functionality, appearance, availability, and price, but also appreciations by its shoppers. Furthermore, the sporadic nature of the Grid and its measured values and the possibility to integrate ad hoc services [2] in a Grid environment of which no historical data is available poses a severe limitation on prediction services.

This motivated us to design a reputation service for Grids to assist in the selection process for resources while integrating the notion of trust and reputation. Trust is already a critical parameter in the decision making process of several peer-to-peer (P2P) frameworks. Reputation is computed using a trust rating provided by users of services through a feedback mechanism. Reputation-based service and product selection has proved to be a great asset for online sites such as eBay [4] and Amazon [5].

Hence, we propose a sophisticated Grid service that selects through a hierarchical process, sets of resource and service as suitable candidates to fulfill quality of service requirements. This includes the selection of trusted resources that best satisfies application requirements according to a predefined trust metric. Therefore, we propose that our hierarchical resource selection process is augmented by the qualitative and quantitative experiences in regards to previous transactions with resources so we can integrate this experience in future resource selections.

We envision such a reputation system for Grids, in which resources and services are ranked based on the reputation they obtain. Generating a reputation or establishing trust by entities (resources, services, and individuals) in regards to their availability and capability. We believe that such a reputation service framework is of crucial importance for Grid computing to increase reliability, utilization, and popularity. Trust and reputation serve as an important metric to avert the usage of under provisioned and malicious resources with the help of community feedback; they provide the ability to simplify the selection process while focusing first on qualitative concerns.

Consider the example to design a Grid environment that agglomerates expensive and specialized resources including high-performance servers, storage databases, advanced scientific instruments, and sophisticated services to visualize macromolecules [6] or nano-material [7] structures. In these usage scenarios we require the availability of reliable ad hoc Grid services to fulfill the necessary quality of service requirements posed by the secured real-time use. Furthermore, the sporadic and time limited nature of the services and resources used may result in a lack of historical data posing severe limitations on prediction services.

Community based adaptive metrics like trust and reputation serve as building blocks to support our quality of service requirements. It is important to recognize that the self-evaluation of a service must be an integral part of the Grid architecture in order to increase reliability and predictability. Consider the case in which a service claims it will provide a particular level of quality and engages in a service level agreement with another service. Assume, this service fails to deliver the promised agreement. Hence, the request is not fulfilled. Choosing a more reliable service can avoid this problem. We conclude that it is imperative to provide a service that evaluates the promised agreement and is available for future reference. We introduce a new framework and algorithm, called GridEigenTrust.

Our paper is structured as follows. In Section II and III, we define the terms trust and reputation and provide an overview of the existing reputation systems for the Grids and their limitations. In Section IV-A, we present the general requirements of Grid reputation framework and service. In Section IV and V, we propose a new algorithm for managing reputation in Grid-based systems and discuss its underlying architecture. After we provide an overview of other related work we summarize future work and conclude our work.

II. Trust and Reputation

In this section we define the basic terminology that will be used throughout the rest of the paper.

A. Definition: Trust

Trust is an ambiguous concept that defies exact definition. However, a notion of trust can be established with sufficient detail for specific operational purpose. For our proposed framework, we define trust as the underlying principle for a security mechanism applicable in a global context. As such, trust is a mechanism for reducing risk in unknown situations. Hence, trust has an important role as a commodity that enables interactions in an unfamiliar environment while weighing the risks associated with actions performed in that environment.

B. Definition: Reputation

Reputation refers to the value we attribute to a specific entity[1], including agents, services, and persons in the Grid, based on the trust exhibited by it in the past. It reflects the perception that one has of another's intentions and norms. Resource reputation provides a way of assigning quality or value in regards to a resource. If a resource is known to provide certain qualities over a period of time irrespective of its limitations, then it is assumed to have good reputation.

C. Definition: Reputation Service

A reputation service is defined as a secure informative service responsible for maintaining a dynamic and adaptive trust and reputation metric for its community. Grid resources, including services providers and consumers, continuously interact with the reputation service to create a community rating mechanism that co-operatively assists their future decisions based on the overall community experiences.

III. Basis of GridEigenTrust

Before discussing our Grid reputation management framework and the GridEigenTrust algorithm, we provide a short overview of current research efforts that form the basis of our work. The GridEigenTrust algorithm is inherently based on the peer-to-peer (P2P) EigenTrust algorithm [8] and the use of reputation to define evolving and managed trust in Grids through the introduction of global trust [9]. The GridEigenTrust algorithm combines these algorithms making it conducive for a large Grid environment by increasing its scalability.

[1] For simplicity, we refer to a resource, service, and a user as entity in the rest of the paper.

A. *EigenTrust Algorithm for P2P Networks*

A reputation management algorithm for P2P networks, called EigenTrust, is introduced in [8]. Every peer i rates other peers based on the quality of service they provide. Therefore, every peer j with whom i had business with will be rated with a grade s_{ij} ($i \overset{s_{ij}}{\rightarrow} j$). To globalize this algorithm the individual grading scheme is normalized as described in [8]. Hence, for each peer j, the normalized local trust value c_{ij} is defined as follows:

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum\limits_j max(s_{ij}, 0)} \tag{1}$$

The normalized local trust values throughout the P2P domain needs to be aggregated. This procedure can be done by means of a transitive trust mechanism: peer i asks its acquaintances for their opinions about other peers and weighs the opinion by the trust it places in his friends:

$$t_{ij} = \sum_k c_{ik} c_{kj} \tag{2}$$

where t_{ij} represents the trust that peer i puts in peer j based on the opinion of his friends $\{k\}$. The coefficients are assembled into a matrix, $C = [c_{ik}]$, hence the equation (2) is written in matrix notation as shown in equation (3).

$$\vec{T_i} = C^T \vec{c_i} \tag{3}$$

The process of obtaining the trust values of friends is repeated to obtain the transitive closure of the matrix (i.e., $T = (C^T)^2 c_i$ would mean that peer i is asking for opinion of his friends' friends, and $T = (C^T)^3 c_i$ for the opinions of their friends). Therefore, after n iterations, where n is the rank of the matrix, the transitive trust is obtained. Hence, T should converge to the same vector for every peer i. Since C is a row stochastic matrix, its largest eigenvalue is 1. Hence, the principal eigenvector of C^T is computed (i.e. the left eigenvector of C). However, this algorithm converges very fast because of the size of the second eigenvalue as shown in [10].

B. *Managing Reputation in Grid Networks*

In [9], [11] several aspects of trust values are considered as part of the global reputation model. First, the trust values decay with time. Second, trust relationships are based on a weighted combination of the *direct* relationship between domains as well as on the *global* reputation of the domains. Finally, the trust model should stimulate organizations to sanction entities who are not behaving consistently in the Grid environment and who break trust relations.

To simplify the notation of our contributions throughout the paper we follow the notation as introduced in [9], [11].

- Let D_i and D_j denote two domains.
- Let $\Gamma(D_i, D_j, t, c)$ denote a trust relationship based on a specific context c at a given time t of D_i towards D_j.
- Let $\Theta(D_i, D_j, t, c)$ denote a direct relationship for the context c at time t of D_i towards D_j.
- Let $\Omega(D_j, t, c)$ denote the global reputation of D_j for the context c at time t.
- Let $DTT(D_i, D_j, c)$ denote a direct trust table entry of D_i for D_j for context c. It is a table that records the trust value from the last transaction between D_i and D_j.
- Let $\Upsilon(t - t_{ij}, c)$ denote the decay function for specific context c where t is current time and t_{ij} is the time of the last update of DTT or the time of the last transaction between D_i and D_j.

Contexts in Grids can be numerous, varying from executing jobs, storing information, downloading data, and using the network. The main issue in trust management is computing $\Gamma(D_i, D_j, t, c)$. In [9], [11], $\Gamma(D_i, D_j, t, c)$ is computed as the weighted sum of direct relationship between domain and global reputation of the domain.

$$\Gamma(D_i, D_j, t, c) = \alpha \cdot \Theta(D_i, D_j, t, c) + \beta \cdot \Omega(D_j, t, c) \tag{4}$$

where $\alpha, \beta \geq 0$, $\alpha + \beta = 1$.

The direct relationship is affected by the time elapsed between inter-domain contacts, hence

$$\Theta(D_j, t, c) = DTT(D_i, D_j, c) \cdot \Upsilon(t - t_{ij}, c) \tag{5}$$

The global trust for domain D_j is computed as

$$\Omega(D_j, t, c) = \frac{\sum\limits_{k=1}^{n} DTT(D_k, D_j, c) \cdot R(D_k, D_j) \cdot \Upsilon(t - t_{kj}, c)}{\sum\limits_{k=1}^{n} (D_k)} \tag{6}$$

where $R(D_k, D_j)$ is the recommender's trust level.

Since reputation is primarily based on what domains say about another domain, the recommender's trust factor $R(D_k, D_j)$ is introduced to prevent cheating through collusions among a group of domains. Hence, $R(D_k, D_j)$ is a value between 0 and 1 and will have a higher value if D_k and D_j are unknown or have no prior relationship among each other and a lower value if D_k and D_j are allies or business partners.

IV. GridEigenTrust Framework

Next we introduce more details about our proposed GridEigenTrust framework. We start with providing general requirements, the introduction of the semantic of our framework, and an elaboration of the algorithms enabling the introduction of a Global reputation in Grids.

A. Requirements

Any given reputation framework for the Grid must adhere to a basic set of minimal requirements.

a) Simplicity: The system should adhere to a simple design that enables minimal overhead in terms of computational, infrastructure, storage requirements.

b) Fairness: The framework should be fair while calculating the reputation. Ideally, the reputation of an institution should not calculated within the institution; rather it must be computed by combining independent evaluations from external services reusing the institutions entities.

c) Robustness: The system should not enable advantages for malicious entities with poor reputations to continuously change their identities to obtain new status. To avoid false reporting a mechanism must be provided to evaluate the accuracy of the reported reputation. Additionally, newcomers in the system will be penalized and established entities will be awarded to encourage and improve consistent good behavior over time.

d) Scalability: The system should be scalable to assist a large community. For example, scalability within the Grid environment should be increased by interacting with additional information services to, for example, maintain load balancing. If the resource selection decisions are contingent only on the reputation severe load imbalance can occur in a large-scale Grid with some dominant resources.

B. GridEigenTrust Framework Semantics

The approach discussed in Section III-B has several limitations. First, under the assumption that we have several domains it is costly to compute the global trust (Equation 6) as we will have to consider all domains in the network for increased accuracy.

Hence, its scalability is limited. To improve scalability, one can compute the global trust among a set of neighbors; however, this would represent a global trust between neighbors but local trust. Second, the authors suggest in their study limiting the number of contexts on. Specifically, the authors reduced the number of contexts in the study to only three: printing, storage, and computing. However, in Grid environments we deal with many more contexts than just printing, storage, and computing. An example would be the evaluation of trust and reputation for network characteristics which is an essential part of any Grid infrastructure.

The eigenvalue approach chosen in III-A is explicitly designed for P2P networks. It has not been applied to the underlying architecture of Grids that introduce virtual organizations providing an obvious classification of resources, users, and their reputation.

Hence, we design a new algorithm that overcomes the limitations of these two approaches. We apply the EigenTrust algorithm explained in III-A to address the problems of scalability and multiple contexts while at the same time introducing a global trust value based on the ability of institutions to maintain a trusted Grid environment and provide the high-performance community with reputation services.

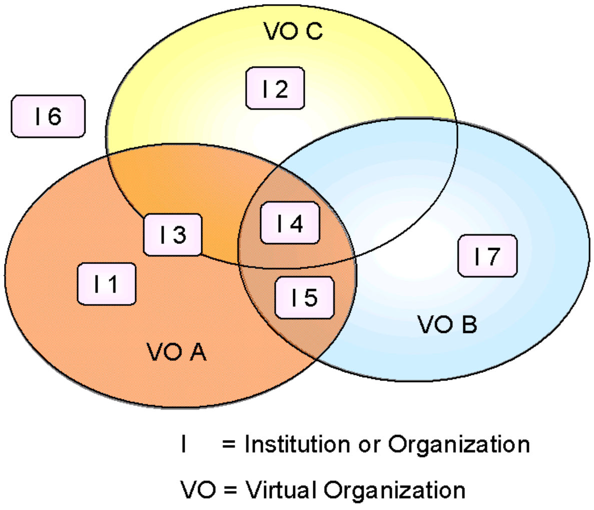

Fig. 1. Institutions contribute in various ways their resources and services to possibly various virtual organizations.

To apply these frameworks to community Grids [2] it is important to revisit in more detail the role of virtual organizations and institutions participation in creating them. As shared resources in a virtual organization are contributed by various institutions it is important to recognize the need of an elaborate reputation service network that deals with the fact that resources can be part of multiple domains and VOs. The different cases are depicted in Figure 1. Here, the institutions I_1, I_7 and I_2 are a part of virtual organizations A, B and C respectively, whereas one part of I_3 belongs to VO A and the other part belongs to VO C. Institution I_6 does not belong to any of these virtual organizations. Considering these various possibilities, the management of reputation in Grids becomes quite complex.

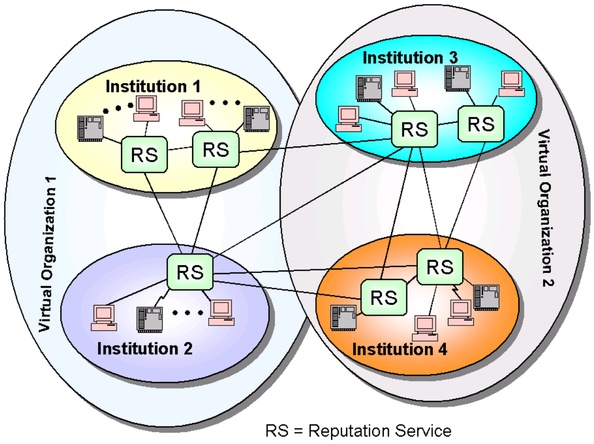

Fig. 2. Example of a distribution of reputation management framework based on reputation services in a Grid.

We address the complexity by introducing a set of reputation services that may even be arranged in hierarchies. To illustrate this point, let us consider the scenario shown in Figure 2.

In this scenario, two VOs are depicted containing two institutions each. Each institution has a set of entities, specifically physical resources, services, and users. Hence, we have introduced an implicit hierarchy based on entities, institutions, and virtual organizations. We assign a reputation to the entities in the lowest level. Based on the reputation of the entities the reputation of the institution gets updated. Finally we compute the reputation of a virtual organization by using the reputation values of all the institutions that belong to the virtual organization. Our reputation service can be reused and integrated in each level of the hierarchy.

The number of reputation services needed for a virtual organization or institution may vary based on its implicit size determined by the entities and the hierarchy they define. Each reputation service is responsible for a subset of entities within the hierarchy. The reputation services compute the reputation in a collaborative, but distributed fashion. It will even be possible to distribute previous reputation values from entities in the network in order to increase lookup speeds. In order to calculate and maintain the reputation, each reputation service uses the GridEigenTrust algorithm described in the next section.

C. *GridEigenTrust Algorithm*

To describe our GridEIgenTrust algorithm we reuse the notation used in Section III-B.

First, we establish a trust value for each entity based on various contexts it supports within an institution. Second, we use the term *reliability trust* for referring to a trust value for each institution. *Reliability trust* differs from other context trust due to the fact that it agglomerates several context trust values to a single one. *It reflects a general opinion of the reliability of an institution to provide accurate information on what resources this institution supplies.* Due to this simplification a reliability trust between institutions can be calculated quickly to obtain the global trust.

By combining reliability trust of the institution, which is not present in the previous introduced methods, and the trust level of a entity within institution (for specific context c at time t) we can derive a reliable trust value for the given entity. We apply the eigenvector mathematical model to compute the global reputation of an institution. Currently, we compute the reputation of a virtual organization as weighted sum of the reputations of all institutions that belong to the virtual organization.

D. *Calculating the Trust of Entities*

To describe how an institution maintains trust parameters of its entities, we modify the notation from Section III-B. Since we are calculating trust values *locally*, i.e. within an institution, we omit the first parameter in the function specification, which denotes the entity from which the trust value was obtained.

All entities that use resources or collaborate with users within another institution grade the quality and reliability of the requested entity. When the entity represents a certain resource or service, we usually call this trust value *service reputation*. When entities represent users, this value represents a *user reputation*, trust, or reliability parameter associated with the user. The overall grade of the entity is established as the weighted sum of the previous grade (which decays with time) and the new grade. It is also important to consider how much we trust the institution from which the remote entity (i.e. entity that gives the grade) originates its requests.

If $\Theta_p(D_i, t_i, c)$ is the previous cumulative grade established at time t_i for entity D_i within context c, $G_j(t, c)$ is a new grade given by entity from institution I_j and $T(I_j)$ reliability trust level of institution I_j, overall new cumulative grade $\Theta(D_i, t, c)$ can be calculated as

$$\Theta(D_i, t, c) = \frac{\alpha(c) \cdot \Theta_p(D_i, t_i, c) \cdot \Upsilon(t - t_i) + \beta(c) \cdot T(I_j) \cdot G_j(t, c)}{\alpha(c) + \beta(c)} \tag{7}$$

where $\alpha(c), \beta(c) \geq 0$.

We notice that equation 7 is similar to 5 from Section III-B. However, the parameters $\alpha(c)$ and $\beta(c)$ reflect the context importance of the latest grade the entity received.

If an institution just joined the Grid, the initial trust values will be set to a low initial value since the trust must be earned first. However, if the entity for which we assign the trust is sufficiently similar to others in the already existing Grid, an initial value can be obtained from these already integrated entities. We chose the lowest trust value. However, it will be penalized with a linear correction function.

Let $\Theta_0(D_i, t_0, c)$ denote the initial trust value for an entity D_i within our institution for a context c. Let $\Theta(D_i, t_i, c)$ denote the cumulative reputation value gathered from other entities (defined by equation (7)). Then the initial trust of the entity is the weighted sum between these two values:

$$\Gamma(D_i, t, c) = \frac{\gamma(c) \cdot \Theta_0(D_i, t_0, c) + \delta(c) \cdot \Theta(D_i, t_i, c)}{\gamma(c) + \delta(c)} \tag{8}$$

where $\gamma(c), \delta(c) \geq 0$.

E. Calculating the Reliability Trust between Institutions

The reliability trust of institution I_i toward institution I_j reflects *the opinion of institution I_i about the quality and trustworthiness of information institution I_j supplies.* Therefore, we introduce besides maintaining individual contexts also global context (compare Section III-B). We use a similar notation as we used in the Section III-B but we omit the parameter c. In case we do have a priori knowledge about the initial trust information we assign this value at initialization time of our algorithm.

Let the initial value of trust be represented as $C(I_j)$. *Reliability* trust should be obtained through the weighted sum of direct experience and global trust value of institution I_j.

Direct experience can be calculated in the same way as in equation 7. It is a normalized weighted sum between $C(I_j)$, the cumulative grade from the previous period $\Theta_p(I_i, I_j, t_{ij})$ and the new grade $G(t)$.

Users within institution I_i grade the reputation of a certain entity D_j within institution I_j with grade $\Phi(D_j)$. Also, institution I_j advertises the quality of service of this entity with grade $\Delta(D_j)$. Then, institution I_i will grade reliability of information given by institution I_j with grade $G(t)$. For determining grade $G(t)$ we have three cases:

- If $\Phi \in [\Delta - \epsilon, \Delta - \zeta]$, new grade $G(t)$ is 1.
- If $\Phi > \Delta - \zeta$, new grade $G(t)$ is bigger than 1.
- If $\Phi < \Delta - \epsilon$, new grade $G(t)$ is less than 1, depending on how much the Φ differs from Δ

Direct experience that institution I_i has with I_j at some time t, $\Theta(I_i, I_j, t)$ can be calculated in the same way as in equation 7. It is a normalized weighted sum between $C(I_j)$, cumulative grade from the previous period $\Theta_p(I_i, I_j, t_{ij})$ and the new grade $G(t)$.

$$\Theta(I_i, I_j, t) = \frac{\alpha \cdot C(I_j) + \beta \cdot \Theta_p(I_i, I_j, t_{ij}) \cdot \Upsilon(t - tij) + \gamma \cdot G(t)}{\alpha + \beta + \gamma} \tag{9}$$

where $\alpha, \beta, \gamma \geq 0$.

Global reliability trust of institution I_j, $\Omega(I_j, t)$ can now be calculated with the EigenTrust algorithm explained in the Section III-A. If we replace c_{ij} with $\Theta(I_i, I_j, t)$ in Section III-A, we obtain a matrix $C = [\Theta(I_i, I_j, t)]$, and initial vector $\vec{T_0} = t_0(i)$, $t_0(i) = C(I_i)$. Now we have all the ingredients to apply a power iteration for computing the principal eigenvector of C^T, which represents global reliability trust values for institutions in Grids.

We can summarize the basic steps of the algorithm as follows:

Entity D_i within institution I_1 wants to use entity D_j within institution I_2 in the context c at time t.

- Consider the reliability trust of I_2 computed using the EigenTrust algorithm, $\Omega(I_2, t)$.
- Ask D_i about $\Gamma(D_j, t, c)$, the trust value of institution D_j within institution I_2.
- In calculating the overall trust value for entity D_j, in formula (4) replace $\Omega(D_j, t, c)$ with $\Omega(I_2, t) \cdot \Gamma(D_j, t, c)$
- Compute the overall trust for the entity $\Gamma(D_i, D_j, t, c)$ with formula (4) and (5).

After computing the trust values, we can compare them to suggest the resource with highest reputation. Various modifications, such as the introduction of a statistical selection algorithm based on random variables, are obviously possible.

This combined approach has several advantages. First, this algorithm converges rapidly and introduces less overhead than computing global trust values for individual entities within every context. One of the reasons is the number of values for computation is not too large since we are computing global trust values of institutions through hierarchies, not an overall pool of individual entities. Second, institutions would make an effort to report accurate trust information about their entities since wrong information will be penalized, lowering the global trust of the institution.

V. Reputation Service Architecture

The architecture of an individual reputation service is shown in Figure 3. It consists of a collection manager, calculation manager, data collection manager, and reporter. The collection manager is responsible for evaluating the quality statement describing the requested reputation, and collecting relevant data from the entities such as resources and users. It gives the collected data to the computation manager. The computation manager computes the reputation values of entities based on the context specified and gives the result to the Storage manager that stores the values to maintain a global and historical view. The reporter contacts the storage manager to report the reputation values whenever queried by some entity in the Grid.

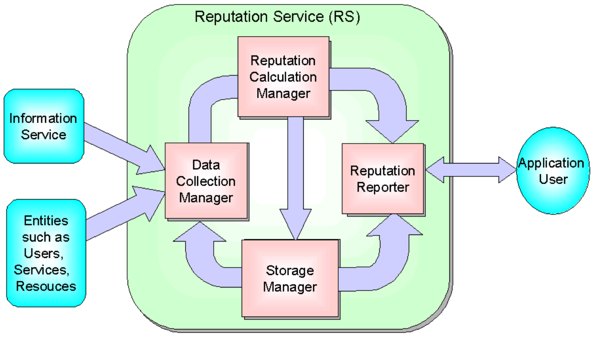

Fig. 3. Architecture of a reputation service.

Hence, when an application submits a request for a service cast in a qualitative statement to the reputation service, the reputation service evaluates the statement and computes the reputation for all the entities providing the required service using the heuristics explained in Section IV-C. It contacts other reputation services if required and returns the information regarding the services and their reputation back to the requester. Hence the requester can decide to select the service by looking at the reputation values. This procedure can be easily modified for enabling and enhancing automating resource selection decisions in the Grid.

VI. Related Work

Reputation has been considered in a wide variety of systems.

- Buyers and Sellers Reputation: The online auction system eBay [4] is an important example of successful reputation management. In eBay's reputation system, buyers and sellers can rate each other after each transaction, and the overall reputation of a participant is the sum of these ratings over the last six months. This system relies on a centralized system to store and manage trust ratings. Furthermore, the information portal c—net maintains also an editors ranking on products and resellers. However individual user responses are not integrated in a correction of the editors ranking.
- Information Ranking: Google employs the principal eigenvector of the matrix to compute the PageRank [12]. PageRank is one of the methods Google uses to determine a page's relevance or importance.
- Trusted Interactions: PeerTrust [13] aims to develop a trust mechanism for system in which peers can quantify and compare the trustworthiness of other peers and perform trusted interactions based on their past interaction histories without trusted third parties. Work includes a trust model and a decentralized and secure trust manager.
- Trust augmentation: The project entitled "Managing Trust Decentralized Applications" [14] aims to provide solutions for decentralized trust management. The main focus is on turning current decentralized information systems into trusted environments in which participants can accurately assess the trustworthiness of their eventual partners in electronic exchanges.

However, these projects do not address the problem of dealing with multiple contexts as we do for the Grid. Unrelated to the above efforts, there are also a few resource management frameworks suggested as part of traditional Grid approaches such as Condor/G [15], Nimrod/G [16] and AppLeS [17]. We envision that our reputation model can be used to enhance these services. The features that distinguish our work from the existing resource brokerage systems are:

- An automated resource discovery based on reputation for information about resource availability at any time.
- A Generic brokerage system that is not coupled to any specific application and that analyzes historic reputation information based on agreement fulfillment.
- Dynamic information gathering through a peer network and management framework while including information about resource availability, global reputation, and the ranking based on reputation.
- Usage policy frameworks for resource providers/administrators as well as users to

enable fine-grained quality of service request specification in regards to reputation.

VII. Conclusion and Future Work

In this paper, we have described a framework for calculating reputation in Grid-based system. The underlaying algorithm to calculate the trust is scalable and robust. It is based on introducing a global trust value that is updated with an eigenvalue based trust calculation algorithm. At present we are in the process of enhancing and evaluating our framework by introducing a variety of reputation measurements that are controlled through adaptive parameters. Such parameters include malicious entities, as well as performance differences while using a variety of update frequencies.

Acknowledgment

This work was supported by the Mathematical, Information, and Computational Science Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38. DARPA, DOE, and NSF support Globus Project research and development. The Java CoG Kit Project is supported by DOE SciDAC and NSF Alliance. We thank Dr. Paul E. Plassmann for his detailed and insightful comments on the paper.

References

[1] I. Foster and C. Kesselman, Eds., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, July 1998.

[2] G. von Laszewski, G. Pieper, and P. Wagstrom, “Gestalt of the Grid,” in *Performance Evaluation and Characterization of Parallel and Distributed Computing Tools*, ser. Series on Parallel and Distributed Computing. Wiley, 2003, (to be published). [Online]. Available: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--gestalt.pdf

[3] G. von Laszewski, S. Fitzgerald, I. Foster, C. Kesselman, W. Smith, and S. Tuecke, “A Directory Service for Configuring High-Performance Distributed Computations,” in *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing*, 5-8 Aug. 1997, pp. 365–375. [Online]. Available: http://www.mcs.anl.gov/~gregor/papers/fitzgerald--hpdc97.pdf

[4] “ebay,” Web page. [Online]. Available: http://www.ebay.com

[5] “amazon,” Web page. [Online]. Available: http://www.amazon.com

[6] G. von Laszewski, M. Westbrook, I. Foster, E. Westbrook, and C. Barnes, “Using Computational Grid Capabilities to Enhance the Ability of an X-Ray Source for Structural Biology,” *Cluster Computing*, vol. 3, no. 3, pp. 187–199, 2000. [Online]. Available: ftp://info.mcs.anl.gov/pub/tech_reports/P785.ps.Z

[7] G. von Laszewski, K. Amin, M. Hategan, and N. J. Zaluzec, “GridAnt: A Client-Controllable Grid Workflow System,” in *37th Hawai’i International Conference on System Science*, Island of Hawaii, Big Island, 5-8 Jan. 2004. [Online]. Available: http://www.mcs.anl.gov/~gregor/papers/vonLaszewski--gridant-hics.pdf

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *Twelfth International World Wide Web Conference, 2003*. Budapest, Hungary: ACM Press, May 20-24 2003. [Online]. Available: citeseer.nj.nec.com/article/kamvar03eigentrust.html

[9] *Evolving and Managing Trust in Grid Computing Systems*. Hotel Fort Garry, Winnipeg, Manitoba, Canada: IEEE Computer Society Press, May 12-15 2002. [Online]. Available: http://www.cs.mcgill.ca/~anrl/PUBS/ccece2002_farag.pdf

[10] T. H. Haveliwala and S. D. Kamvar, “The Second Eigenvalue of the Google Matrix,” Web page. [Online]. Available: http://www.stanford.edu/~sdkamvar/papers/secondeigenvalue.pdf

[11] *Integrating Trust into Grid Resource Management Systems*, The International Association for Computers and Communications. Vancouver, B.C., Canada: IEEE Computer Society Press, Aug. 18-21 2002. [Online]. Available: http://www.cs.umanitoba.ca/~anrl/PUBS/icpp2002_farag.pdf

[12] S. D. Kamvar, T. H. Haveliwala, and G. H. Golub, “Adaptive Methods for the Computation of Page Rank,” Web page. [Online]. Available: http://www.stanford.edu/~sdkamvar/papers/adaptive.pdf

[13] “PeerTrust Overview,” Web page. [Online]. Available: http://www.cc.gatech.edu/projects/disl/PeerTrust/

[14] “Managing Trust in Decentralized Applications,” Web page. [Online]. Available: http://lsirwww.epfl.ch/projects/swiss/trust-project.htm/

[15] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke, “Condor-G: a computation management agent for multi-institutional grids,” in *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium*. San Francisco, CA, USA: IEEE Computer Society Press, Aug. 2001, pp. 55–63.

[16] R. Buyya, D. Abramson, and J. Giddy, “Nimrod/G: An Architecture of a Resource Management and Scheduling System in a global Computational Grid,” in *the 4th International Conference on High-Performance Computing in the Asia-Pacific Region*. IEEE Press, May 2000, pp. 283–289. [Online]. Available: http://www-unix.gridforum.org/mail_archive/perf-wg/pdf00000.pdf

[17] F. Berman and R. Wolski, “The AppLeS Project: A Status Report,” in *The 8th NEC Research Symposium*, May 21-22 1997. [Online]. Available: citeseer.nj.nec.com/berman97apple.html