A reputation algorithm for a self-organizing system based upon resource virtualization.

Conference Paper · January 2008

Source: DBLP

5 authors, including:

Dan C Marinescu
University of Central Florida
175 PUBLICATIONS **1,527** CITATIONS

Chen Yu
University of Central Florida
16 PUBLICATIONS **69** CITATIONS

John P. Morrison
University College Cork
123 PUBLICATIONS **519** CITATIONS

A Reputation Algorithm for a Self-Organizing System Based upon Resource Virtualization

Dan C. Marinescu, Chen Yu, Gabriela M. Marinescu
School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, Fl, 32816
Email: (dcm, yuchen, magda)@cs.ucf.edu

John P. Morrison and Christoffer Norvik
Computer Science Department
University College Cork, Cork, Ireland
Email: (j.morrison, c.norvik)@cs.ucc.ie

October 4, 2007

Abstract

In the service-oriented architecture introduced in [10] the reputation affects the prices producers could ask for their resources, the prices consumers of resources are willing to pay, and the commission charged for every contract. In this paper we introduce the GRB algorithm to quantify the reputations of the participants. The system is self-organizing and this affects the space requirements and the communication complexity of the algorithm.

1 Introduction

Traditionally, a service provider in a service-based distributed system owns, or is in total control, of all resources needed to provide that service. Peer-to-peer systems are challenging this paradigm with varying degrees of success. For example, Skype, an Internet phone service provider, surreptitiously uses selected sites connected to the system as super nodes to route calls; Napster, a file sharing system stores data files on individual sites owned by its subscribers and Video-on-demand attempts to take advantage of data storage on sites scattered around the Internet. The strategy used by these applications and others is motivated by a desire to reduce the cost of providing the service.

Current implementations of the peer-to-peer resource sharing philosophy is plagued by problems such as uncertainty regarding the amount of resources available at any instant of time, asymmetry of the communication bandwidth of home access networks (the download bandwidth is generally much higher than the upload one), and lack of system support at the sites providing a fraction of their resources to the community. As a result, such systems are unable to provide Quality of Service (QoS) guarantees or ensure fairness, are vulnerable to malicious attacks, prone to catastrophic failures, as the recent collapse of Skype shows, do not provide security guarantees either to the supplier of resources, or to the user of the system (often embodied by the same individual), are restricted to a few services with a simple user interface, and are limited in their ability to use a wide spectrum of resources.

The service-oriented architecture succinctly presented in Section 2 is based upon resource virtualization and needs a mechanism to assess attributes such as availability, reliability, Quality of Service (QoS), fairness, and the security risks posed to the principals involved in different types of transactions [10]. In this paper we introduce algorithms to quantify such attributes and model our approach on *reputation services* used by different Internet sites.

The centerpiece of a *reputation service* is a collaborative filtering algorithm to determine the ratings of a collection of principals based upon objective measurements, as well as opinions that they hold about each other. The quality and complexity of the filtering algorithm varies widely depending upon the circumstances and the context of its use.

Reputation is based on objective measurements, claims (verified or not), transactions, ratings, and endorsements. Reputation services are already in place at social networks (Slashdot, Kuro5hin), E-commerce sites (eBay, Amazon), product review sites (such as, Epinions, BizRate, Edmunds), expert sites (e.g., AskMe, Advogado), and search engines (e.g., Google) [8]. For example, the auction site `eBay.com` invites the parties involved in a transaction to rate their each other and these ratings are open to review by everyone. The online store `amazon.com` encourages individuals to review products and allows potential customers to vote if a review was helpful or not. Based upon the vote the reviewers themselves are classified in the Top 100, Top 10, etc. Another example of a reputation algorithm is PageRank used by `google.com`: "PageRank relies on the unique democratic nature of the Web by using its vast link structure as an indicator of an individual page value. In essence, Google interprets a link from Page A to Page B as a vote by Page A, for Page B."

Trust and reputation services are plagued by a host of problems such as: low incentives for providing ratings, bias towards positive ratings, unfair ratings, change of identities, discrimination, quality variation over time, and ballot stuffing. Some of these factors are self explanatory, others require some comments. Ratings are provided after the completion of a transaction, thus, the parties involved may not benefit directly and tend to ignore the need to provide ratings. The rating may only be provided when negative opinions are expressed and statistics show that negative opinions are rarely expressed; for example on eBay less than 2.5% of the ratings provided by either buyers or sellers were negative.

Reputation is based on identity and a new identities can be acquired with ease; a low ranked service may shut down and reappear under a different name. Reputation is a currency and untrustworthy principals attempt to undermine the system. For example, *ballot stuffing* is a mechanism to bolster the reputation of one or more principals; even when rating is accepted only from the participants in a transaction, a group of dishonest principals may conduct circular transactions to bolster their collective reputation and then use it to conduct fraudulent transactions. The filtering algorithms should identify and defeat such attempts.

Trust and reputation systems used in practice are generally simple but are affected by the negative factors mentioned above; the ones proposed by academia have a sound mathematical basis, e.g., an algebra for trust certification [6], or a probabilistic foundation [7], but are cumbersome and require parameters that are not readily available.

In this paper we propose an algorithm to circumvent the problems of other reputation systems. In our service-oriented system the reputation affects the prices that producers could ask for their resources, the prices that consumers are willing to pay, and the commission charged for every contract. Therefore we attempt to extend the weight of reputation based upon direct measurements, reduce the possibility of cheating, discourage contract renegotiation, and heavily penalize breach of contract.

2 A Service-Oriented System Based upon Resource Virtualization

To understand the requirements of the algorithm introduced in this paper we outline the context of its use. The resource management architecture of the service-oriented system we propose is inspired by economic systems where a large number of individuals provide financial resources needed by a company by buying shares of that company; they are rewarded by dividends and the expectation that the company will do well and the price of the shares will increase.

We assume a large population of principals who have an excess capacity of perishable computing resources such as: CPU cycles, main memory, and secondary storage and are willing to contract out their excess capacity to a Central Market Authority (CMA). The CMA bundles these resources based upon a set of attributes such as: quantity, price, QoS, security, location, and network access in order to match the needs of a large population of resource consumers. Though immaterial for the present discussion, we note that the resource consumers may be providers of services that may in turn be requested by some of the resource providers.

This process of bundling resources is called *resource virtualization.* Resource virtualization allows transparent access to resources; in our context this means that the resources available to a service provider are actually scattered throughout the system, the same way the pages of a virtual address space are scattered throughout the physical memory. This analogy, intuitive as it may be, has serious limitations. First, the *virtual memory management subsystem* is in total control of the resources it is supposed to manage, the physical memory, while a *resource management agent* must deal with the fact that some remote resources become suddenly unavailable and a certain degree of replication is always necessary. Thus, the virtual memory management subsystem could make an informed decision to ensure optimality, for example it may suspend a process that causes memory thrashing, while the virtual resource manager does not have the luxury to make optimal decisions due to the dynamics of resource availability. Second, the virtual memory management controls a unique resource, the physical memory, while a resource management agent has to acquire CPU cycles, primary memory, secondary storage, bandwidth, special hardware, and so on. Obviously this complicates the acquisition algorithms and scheduling policies. Last, but not least, the memory management sub-system is shared among all processes running at a given site, while a service provider may require a hierarchy of the resource management agents, because individual clients may have different security, fault-tolerance, and QoS requirements.

A transaction in our system is a complex affair because it involves multiple parties and several contracts; the CMA has separate contracts with a number of resource providers and typically with one consumer. The reputation of the principals involved in such a complex transaction is critical if we require the bundling mechanism to appropriately match the needs of the service providers, i.e. the consumers of resources, with the providers of those resources. Moreover, the reputation affects the price that producers can ask for their resources, the price consumers are willing to pay, and the commission charged for every contract.

The system as described is not scalable if the Central Market Authority is monolithic. We envision that groups of principals, both producers and consumers will eventually form Virtual Organizations (VOs) based on similar attributes such as: interest, location, and security. Each virtual organization may develop its own Local Market Authority, (LMA) so that a large fraction of the transactions between the producers and consumers can be satisfied within the virtual organization. While most of the transactions are carried out within one virtual organization, some may require either the intervention of the CMA and/or multiple LMAs. To maintain the uniformity of notations we refer to CMA as MA^0 and to LMA^k as $MA^k, 1 \leq k \leq K$.

3 The Basic Reputation Algorithm

In this section we describe the algorithm used to evaluate the reputation of principals under the control of one market authority, be it the central or any local authority. We assume initially that there are N principals thus the CMA maintains a reputation table with N entries. At time t there are $N(t) >> N$ principals and the corresponding entries are distributed among the CMA and K $LMAs$; the CMA has $N^0(t)$ principals as members and LMA^i has $N^i(t), 1 \leq i \leq K$ members. Thus:

$$N(t) = N^0(t) + N^1(t) + N^2(t) + \ldots + N^K(t).$$

Each principal is a member of one organization called *home*; each contract is managed by a single market authority. It is possible that producers belonging to different virtual organizations provide resources for one consumer; in this case the contract is managed by the market authority of the consumer. It is also possible that resources provided by one producer are fragments and participate in multiple contracts, each managed by the market authority of each consumer.

The algorithm we propose takes advantage of the special features of the service-oriented architecture in Section 2 to avoid some of the drawbacks of similar algorithms. The main ideas of the Gentle Reputation Buildup (GRB) algorithm is to allow each principal to enhance its reputation gradually in the absence of negative feedback and to heavily penalize reputations in the light of negative feedback.

The upgrade of the reputation is automatic upon the termination of a contract unless a negative feedback from either the principals involved, or the controlling market authority is generated. In addition to the negative feedback the controlling marketing authority penalizes the termination of a contract or the renegotiation of a contract by fining the principal involved. A principal providing negative feedback is charged a small processing fee to discourage gratuitous complaints.

Let $\rho_i^k(n)$ be the reputation of principal i evaluated by market authority $k, 0 \leq k \leq K$ after the completion of n contracts involving the principal i. Initially $\rho_i^k(0) = 0$. In the absence of negative feedback the reputation increases exponentially at first and subsequently experiences a linear increase; given the positive real numbers $\kappa_1, \kappa_2, \lambda$ and ρ_{max} the reputation is:

$$\rho_i^k(n) = \begin{cases} \kappa_1 2^n & 1 \leq n \leq \lambda \\ \min\left[(\kappa_1 2^\lambda + \kappa_2 n), \rho_{max}\right] & \lambda < n \end{cases}$$

As we can see there is a threshold ρ_λ and the reputation increases exponentially until this threshold is reached and then it increases linearly. A principal reaches the highest reputation after completing $n' = \lambda + (\rho_{max} - 2^\lambda)/\kappa_2$ contracts without receiving negative feedback. For example Figure 1 (a) shows the case when $\kappa_1 = 2^{-9}, \kappa_2 = 0.5, \lambda = 10, \rho_{max} = 10$; in this case $\rho_\lambda = 2$ and $n' = 10 + (10 - 2)/0.5 = 26$.

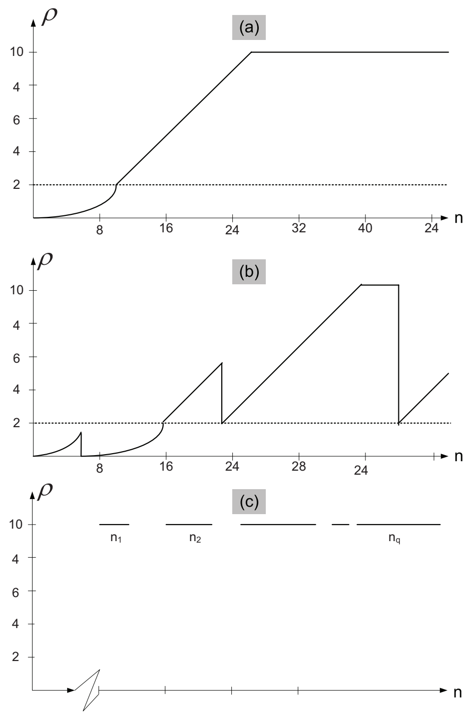

Figure 1: (a) In absence of negative feedback the reputation increases exponentially at first, then experiences a linear increase. (b) Negative feedback drastically lowers the reputation. (c) The segments used to compute the reputation consistency.

Negative feedback, Figure 1 (b), lowers the reputation from its current value $\rho_i^k(n)$ according to the following rules:

$$\rho_i^k(n+1) = \begin{cases} 0 & \text{if} \quad 0 < \rho_i^k(n) \leq \kappa_1 2^\lambda \\ \kappa_1 2^\lambda & \text{if} \quad \kappa_1 2^\lambda > \rho_i^k(n) \leq \rho_{max}. \end{cases}$$

The negative feedback lowers the reputation to 0 if it is below the threshold ρ_λ. The reputation drops to ρ_λ any time negative feedback is received after this threshold is reached.

Another useful metric is the *reputation consistency*, which is defined as follows: assume that principal i maintains the highest reputation for n contracts. There are q periods of duration $n_i^1, n_i^2, \ldots, n_i^q$, expressed as the number of contracts in each period, when $\rho_i^k = \rho_{max}$, Figure 1 (c). Then the probability that the principal i is able to maintain the highest reputation for n_j consecutive contracts is:

$$p_i^j = n_i^j / \sum_{k=1}^{q} n_i^k$$

and the *reputation consistency* of principal i evaluated by a market authority $k, 0 \leq k \leq K$ is:

$$\theta_i^k = -\sum_{j=1}^{q} p_i^j \log p_i^j.$$

θ_i^k allows us to discriminate based upon reputation consistency. For example, let us assume that all periods have the same duration τ and $n = q\tau$. Then $p_j = \tau/n = 1/q, \;\; 1 \leq j \leq q$ and:

$$\theta_i^k = q(1/q)\log q = \log q = \log n - \log \tau.$$

It is easy to see that in this case:

$$0 \leq \theta_i^k \leq \log n.$$

The more consistent the reputation, the closer θ_i^k is to 0; when $\tau = n$ then $\theta_i^k = 0$; when $\tau = 4$ then $\theta_i^k = \log n - 2$; when $\tau = 1$ then $\theta_i^k = \log n$.

The characterization of the reputation of principal i by the pair $(\rho_i^k(n), \theta_i^k)$ has a number of distinct advantages:

(i) it mimics social reputation, it builds up slowly and it is drastically affected by negative feedback;

(ii) it is easy to compute, the function increases monotonically up to the maximum value in absence of negative feedback;

(iii) rewards consistent "good" behavior;

(iv) discourages most forms of cheating, e.g., a principal with low reputation does not gain by changing its identity.

Principals are involved in contracts evaluated by multiple market authorities; the reputation of a principal computed by the home market authority has to be reconciled with its reputation computed by all secondary market authorities. Call $0 \leq \eta^{(k,j)} \leq 1, \;\; 0 \leq k \leq K, \;\; 0 \leq j \leq K, \;\; j \neq k$ the level of trust of MA^k in MA^j. Then the system is characterized by a *mutual level of trust*:

$$\bar{\zeta} = \frac{1}{(K+1)^2} \sum_{k=0}^{K+1} \sum_{j=0}^{K+1} \eta^{(k,j)}, \;\; 0 \leq \bar{\zeta} \leq 1.$$

The effect of MA^j upon the reputation $\rho_i^k(n)$ is then:

$$\omega^{(i,j,k)} = f(\rho_i^j(n), \eta^{(k,j)}/\bar{\zeta}.$$

Then the overall reputation of principal i evaluated by its home market authority MA^k is:

$$\hat{\rho}_i^k(n) = \rho_i^k(n) g(\omega^{(i,j,k)}), \;\; 0 \leq j \leq K, \; j \neq k.$$

The number of principals is expected to increase in time; moreover, the principals may switch from one virtual organization to another based upon an *affinity*, given by the closeness of some attributes. The system we propose is self-organizing [10] and in the next section we discuss the effect of self-organization upon the space and communication requirements of the reputation algorithm.

4 Self-Organization and its Impact upon the Space and Communication Requirements of the Reputation Algorithm

Complex systems exhibit a high degree of self-organization. Self-organization can be best explained in terms of the network model of the system when the vertices represent the principals and the links represent the relationships among them. Self-organization implies that the probability $P(m)$ decays as a power law when a vertex interacts with m other vertices , $P(m) \approx m^{-\gamma}$(regardless of the type and function of the system, the identity of its constituents and the relationships between them). This implies that large systems self-organize into a scale-free state [1, 2, 3, 5].

Empirical data available for social networks, power grids, the Web, the citation of scientific papers confirm this trend. As an example of a social network consider the collaborative graph of movie actors where links are present if two actors were ever cast in the same movie; in this case $\gamma \approx 2.3$. The power grid of Western US has some 5000 vertices representing power generating stations and in this case $\gamma \approx 4$. For the World Wide Web the exponent $\gamma \approx 2.1$; this means that the probability that k pages point to one page is $P(m) \approx m^{-2.1}$ [3]. Recent studies indicate that for scientific papers $\gamma \approx 3$. The larger the network the closer is the distribution approximated with a power law with $\gamma \approx 3$ [1].

The empirical data from the study of large networks are in disagreement with the Erdös–Réni or the Watts-Strogatz models of random graphs [2, 4] which assume that the probability that two vertices are connected is random and uniform. Moreover, these models ignore the fact that the systems encountered in practice continuously expand by addition of new principals connected to the ones already in the system.

The vertices of the network model of the system discussed in [10] are the principals, whether producers or consumers; a link between two vertices i and j exists at time t if at any time prior to t the two principals have been involved in the same transaction. The space complexity of the algorithm is related to the average number of principals in one virtual organization and the communication complexity is related to the number of virtual organizations in place at time t. If we assume a very large number of nodes we expect $\gamma \approx 3$.

One possible strategy for self-organization is to allow the most popular principals to establish their own virtual organizations and then to attract to each VO those principals who have the largest number of transactions with a leader. While an exact analysis is rather intricate, a qualitative analysis can be conducted based upon the basic properties of scale-free networks. For example, if a leader is a principal with at least μ links, then the expected number of leaders, thus $\nu(t)$, the expected number of Virtual Organizations, and $\bar{N}^k(t)$, the expected number of principals in one virtual organization, at time t, are respectively:

$$\nu(t) \approx N(t)P(\mu) = N(t)\mu^{-3} \quad \text{and} \quad \bar{N}^k(t) = \frac{N(t)}{\nu(t)}$$

When $N(t) = 10^9$ and $\mu = 10$ then $\nu \approx 10^9 \times 10^{-3} = 10^6$, and $\bar{N}^k(t) \approx 10^9/10^6 = 10^3$. If we raise the bar and $\mu = 20$ then $\nu \approx 10^9 \times 20^{-3} \approx 10^5$, and $\bar{N}^k(t) \approx 10^9/10^5 = 10^4$.

We know that a set on N elements can be partitioned into ν non-empty sets in a number of ways given by the Stirling number of the second kind:

$$S(N,\nu) = \frac{1}{\nu!}\sum_{i=0}^{\nu}(-1)^i \binom{\nu}{i}(\nu - i)^N.$$

The Strirling numbers of the second kind for $1 \leq N \leq 5$ and $1 \leq \nu \leq N$ are presented in the table below:

$$\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 3 & & 1 & & \\
 & 1 & & 7 & & 6 & & & \\
1 & & 15 & & 25 & & 10 & & 1
\end{array}$$

When $N = 10^9$ and $\nu = 10^6$ then $S(N,\nu)$ is a very large number and among all possible partitions we should select the ones that minimize the number of out-partition links.

5 Summary

In this paper we introduce the GRB reputations algorithm and provide a qualitative analysis of its space and time requirements.

Acknowledgement

One of the authors (dcm) expresses his thanks to the Science foundation of Ireland for the support provided by a Walton award.

References

[1] A-L. Barabási and R. Albert. "Emergence of Scaling in Random Networks." *Science*, 286:509–512, 1999.

[2] A-L. Barabási, R. Albert, and H. Jeong. "Mean-field Theory for Scale-Free Random Networks." *Physica A*, 272:173–187, 1999.

[3] A-L. Barabási, R. Albert, and H. Jeong. "Scale-Free Theory of Random Networks; the Topology of World Wide Web" *Physica A*, 281:69–77, 2000.

[4] B. Bollobás. *Random Graphs.* Academic Press, London, 1985.

[5] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnàdi. "The Degree Sequence of a Scale-Free Random Graph Process." *Random Structures and Algorithms,* 18(3):279–290, 2001.

[6] A. Jøsang. "An Algebra for Assesing Trust in Certification Chains." Proc. NDSS99, Network and Distributed Systems Security Symp. San Diego, pp. 1-10, The Internet Society, 1999.

[7] A. Jøsang, R. Ismail. "The Beta Reputation System." Proc. 15th Electronic Commerce Conference, pp, 1- 14, Bled, June, 2002.

[8] A. Jøsang, R. Ismail, C. Boyd. "A Survey of Trust and Reputation Systems for Online Service Provision." *Decision Support Systems*, 43(2):618-644, 2007.

[9] D. Lebedeev. "Ad hoc Networks: Study of Protocol Behavior." *Ph.D. Dissertation* Ècole Polytechnique, Paris, 2006.

[10] D. C. Marinescu and J. P. Morrison. "The Pleiades: a Self-Organizingq Service-Based Architecture," 2007 (in preparation).