A Reliability-based Trust Management Mechanism for Cloud Services

Wenjuan Fan1*,2 and Harry Perros2

1School of Management, Hefei University of Technology, Hefei, Anhui, China
wan3@ncsu.edu
2Computer Science Department, NC State University, Raleigh, NC, USA
hperros@gmail.com

Abstract:
With the increasing demand for cloud services, trust management has become a challenging and important issue in a cloud computing environment. In a trust management mechanism, trust feedback is used to derive trust evaluation results. However, the reliability of the trust feedback from cloud service users needs to be considered, because unreliable trust feedback can produce wrong trust results. In this paper, we firstly propose a trust management framework in cloud computing environments, and introduce an effective reliability-based filtering mechanism to ensure the reliability of trust feedback for cloud computing services. The filtering mechanism filters unreliable trust feedback based on two important factors: familiarity and consistency, and the combination of these two factors falls to the integration of endogenous and exogenous methods to filter out unreliable ratings. Our experiments show that our proposed reliability-based trust management mechanism is effective.

Key Words: Trust, reliability, familiarity, consistency, trust management, cloud services

1. Introduction

Cloud computing is a gradually maturing information service paradigm where users get access to a cloud through a variety of devices and are allocated virtualized resources delivered in the form of “pay as you go” services, just like using utilities such as electricity and water. The increasing demand for cloud-based services has given rise to the need for an effective trust management mechanism, which can provide useful and reliable trust decisions.

The challenge of trusting cloud computing does not lie entirely in the technology itself. Despite the good features of low investment, flexibility and fast deployment, one of the biggest obstacles from choosing a cloud service is the issue of trust (Tian et al. 2010). The lack of customers’ trust stems from a lack of transparency, a loss of control over data assets, and unclear security assurances (Khaled and Qutaibah, 2010). For potential users, there are risks that service providers are not trustworthy and reliable to provide satisfactory services. So with respect to a cloud environment, trust is an important issue that needs to be considered.

People trust a service or a service provider for the following three reasons: 1. They have a satisfactory direct interaction with a service provider and a service. 2. The service provider has a high reputation spread through word of mouth. 3. There is a good recommendation from someone familiar or trusted parties about a certain service provider or a service. In addition, various factors and evidence have different effects on the trust, and some of them carry more weight than others. Obviously, direct interactions will affect the trust opinion of users most. However, there is not always a direct experience between a user and a service provider. This is the case, for instance, when a user uses a service for the first time. In this case, the user has to rely on the trust feedback from other users who interacted with the service before, in order to decide whether to use the service or not.

However, the historical trust feedback for service may not be reliable, and considering all users’ trust feedback to infer trust maybe misleading. This is because, some of the feedback may be from biased users either in favor or against the service in question. This dishonest positive or negative feedback has to be filtered out in order to estimate the trust correctly. In this paper, we propose to do this by calculating the reliability of the users’ trust feedback using the following two metrics that reflect the behavior of a user, namely, consistency and familiarity. These two metrics are calculated using the trust feedback values of each user and the times instances at which the user used the service in question. The reliability of the trust feedback values of a user is then calculated by multiplying these two metrics, and it is used as a weight function for each user’s trust feedback values in order to calculate the final trust value.

The structure of this paper is as follows. Section 2 gives the background and related work. Section 3 gives a short illustration of the architecture of our proposed trust management mechanism, and in Section 4 we describe the three-step trust management mechanism. In Section 5 we provide simulation results in order to demonstrate how well the proposed method tackles dishonest trust feedbacks. Finally, the conclusions are given in
2. Background and Related Work

Trust plays an important role in E-commerce, P2P networks, and information filtering (Song, 2005). However, trust is one of the most complex relationships between entities, because it is abstract, unstable and difficult to measure and manage. The trust value or the trust degree reflects the extent to which an entity trusts another. Trust values can be either discrete, e.g., -1, 0, 1, or continuous, e.g., [-1, 1]. Trust values may be effective and applicable within a certain context while ineffective in others.

It has been shown that modeling trust and reputation can be used effective in improving security, support decision-making and entity collaboration. In earlier papers in the literature, trust was often used to enhance security, such as authentication, access control, and encryption, and it was part of a security management system (Rasmussen and Jansson, 1996). This approach on trust is known as “soft security” and it is concerned with applications of objective or rational trust. However, this approach is not sufficient for most distributed systems, such as, e-commerce, ad hoc network, wireless sensor network, and P2P systems, which need social collaboration or subjective decision-making. Most recent work has focused on these distributed systems. Trust for distributed systems should be based as much as possible on trust evidence in order to give the most reliable trust decision (Ren et al. 2004). A distributed trust model was proposed that uses a recommendation protocol to exchange trust-related information in a decentralized fashion (Abdul-Rahman, Hatles, 1997).

Many trust management systems propose a distributed trust model that enables a subset of the nodes to evaluate the behavior of neighboring nodes and make decisions about them (Lopez et al., 2010). Zhou and Haas (1999) introduced a key management system based on threshold cryptography for ad hoc networks. Yao et al. (2005) presented a security framework based on trust management system for sensor networks. This model enables recommendation-based trust and trust-based recommendation. Liu and Issamy (2004) proposed a reputation mechanism for MANET which is able to distinguish truth-telling and lying nodes, with a view to preventing defamation. Theodorakopoulos and Barasa (2004) proposed a trust evaluation mechanism for MANETs which can be used in a decentralized infrastructure. In this mechanism, trust evaluation is modeled as a shortest path problem whereby the trust path from trustor to trustee is modeled as a weighted directed graph, in which vertices represent nodes and weighted edges indicates opinions of one node for its each adjacent node. Based on semiring theory, indirect trust relation between trustor and trustee can be established without previous direct interaction. Gupta et al. (2003) proposed a partially distributed reputation system for P2P networks which includes a central reputation computing agent trusted by all other nodes. Wang and Akhiro (2010) proposed a social-network based reputation ranking algorithm to infer reputation ranks. It includes three procedures: (1) the propagation of Poisoned Water (PW); (2) the determination of adaptive Spreading Factor (SF) from PW level and (3) the enhanced group-based reputation ranking algorithm with adaptive SF. Li et al. (2011) proposed a trust model for large-scale P2P computing, in which multiple factors are incorporated to reflect the complexity of trust, and the weights are dynamically assigned using a combination of weighted moving average and ordered weighted averaging. Wang and Wu (2011) proposed a multi-dimensional evidence-based trust management system with multi-paths to conduct trust computation on any arbitrarily complex trusted graph. This is an innovative approach on the three-tier trust computation, i.e., node tier, path tier, and graph tier.

Trust is an essential part of a successful e-commerce business. Successful e-commerce web sites enjoy consumers’ trust and lower their risk perception though effective trust management (Brian J. Corbitt et al., 2003). Probably, the most well known reputation system in e-commerce is the one used by eBay. It is an example of a global reputation system, whereby the reputation rating of a peer is calculated after each transaction as the sum of all the ratings posted for this peer. However, there are several shortcomings in this system. Firstly, there is no mechanism to filter unfair ratings, and secondly, the ratings from different individuals are weighted equally.

So far, there is a limited number of papers on trust management for cloud computing (Li and Ping, 2009), and most of them refer to grid computing, P2P computing and other distributed computing paradigms. Tian et al. (2010) discussed user behavior trust in cloud computing. The basic idea is “divide and treat” based on a hierarchically structured model that decomposes the user behavior trust into small sub-trusts, and each sub-trust (ST) is further subdivided into smaller data units. The results from each level are then aggregated to construct the user behavior trust. In order to enhance the security level of cloud computing, many researchers introduce TPM (Trusted Platform Module) into the cloud computing system (Krauthiem et al., 2010; Hiroyuki et al., 2010; Shen et al., 2010; Li et al., 2010). However, there is a lack of trust models for service providers and users to establish trust reliably. Different from the previous work which mainly focused on the computational trust models,
in this paper we concentrate on how to filter unreliable trust feedback by users so that make cloud service trust management more robust.

3. A framework for the proposed trust system

In e-commerce or e-marketing, item ratings provided by different users are collected by web systems, such as, Amazon and MovieLens (Zheng et al., 2009). However, in the field of cloud computing services, it is difficult to acquire and collect service trust feedback. This is because cloud services are usually owned by different cloud service vendors and the users are thus isolated from each other. As a result, the cloud service trust feedback for services provided by a cloud service provider, cannot be shared by other users on a global basis. Without sufficient cloud service trust feedback information, it is difficult to produce correct and reliable trust decision on cloud service providers. This problem can be avoided using a cloud service trust management system architecture such as the one shown in Figure 1, which enables the collection, storage, extraction, and training of service trust feedback information.

3.1 Proposed trust management system

As shown in Figure 1, this system is divided into two domains: service provider domain and service user domain. In the service provider domain, this system mainly performs the function of connecting the portals of different cloud service providers. The service transactions through the portal of trust management system can be observed and reported by the trust management system, which are used for generating trust results on cloud services. In service user domain, it mainly performs the function of collecting cloud service trust feedback from distributed service users, searching reliable users, deriving the real time trust results on the cloud services.

Based on the trust management system, service providers and users can establish trust relationship with each other according to their specific requirements and rules. Besides, they can also make important trust-related decisions such as service selection and user authority classification. In conclusion, this system provides a publishing platform of service and trust management information. In addition, it selects, filters, judges and aggregates the information so as to connect service users and providers with a view to enhancing trust relation.

Service providers and users should enroll into the system through their respective entries for service provider domain and user domain in order to obtain their provider ID and user ID respectively. Service providers should provide their provider ID, property (individual or organization etc), scale (small, mediate, or large), and other information. Also, for each service, the service provider should provide the SLA, which should include performance guarantees, QoS and cost, the service functions, and basic guarantees on non-functional properties of the services. Users will deem a service provider as trustworthy if its services satisfy the advertised SLA and other specifications.

For users, the system will also create a table of their basic information after they enroll in the system, where they can publish their required information about services. After using a service, they are asked to input their trust feedback for the service. Unlike the [0, 1] trust metric that is typically used in the literature, we use the “general trustworthiness” trust feedback given by five grades, i.e., [1, 5], with 1 being the most untrustworthy, and 5 the most trustworthy.

Figure 1 shows the main components of the proposed trust management system for cloud services, which will be introduced as follows:

In service user domain:

User membership management: this part manages the user-cloud interface by cloud portals, through which users can enroll and onboard into the trust management system, and also has their own identification.

User request management: this layer mainly manages the users’ service requests to the cloud, such as, publishing, classifying, selecting, and recommending services to the users.

Trust feedback management: in this layer, the trust management system will allow users to give their trust feedback to the experienced cloud services, such that evaluate and establish trust relationship to them.

In service provider domain:

Provider membership management: cloud service providers can also enroll into the trust management system through a provider portal/hub, and publish their cloud services in to the system.

Cloud service management: in this layer, the trust management system will monitor the interactions performance such as the network workload, throughout, responsive time, etc.

Trust management: the cloud service provider can differentiate the malicious users from reliable users based on their behavior, such that use different security policies to establish trust relationship with different users.
3.2 Malicious attack model

We assume two main malicious attack models in this paper, i.e., unfairly positive feedback and unfairly negative feedback. 

unfairly positive feedback: This attack happens when the malicious cloud users try to increase their trust feedback assessments unreasonably so that achieve some interests, for example, when they are collusive with the target cloud service provider. 

unfairly negative feedback: This attack happens when the malicious cloud users attempt to downgrade their trust feedback assessments so that considered as the opposite of the Self-promoting attack that happens when the malicious cloud consumers try to decrease the trust results of certain cloud service.

Service users can give trust feedbacks for a certain cloud service or send a query to the trust management system regarding a certain cloud service. In the following sections, we will focus on introducing our design of the mechanism of filtering the unfairly feedback to make trust management more robust.

3.3 Trust Decision-making Process

In this paper, we propose a trust management mechanism that measures the reliability of the trust feedback submitted by users. We assume that the set of service providers (SPs) that can provide a particular service with the necessary functions can be composed using a matching algorithm, which is beyond the scope of this paper. Given such as a set of providers we use the trust feedback of users who have interacted with these SPs before and have returned their trust feedback on the specific service, to calculate their reliability based on the consistency of their historical feedback and familiarity with the SPs. The reliability of the users’ trust feedback is significant for making a trust decision on cloud service. The unreliable users are filtered out based on a threshold reliability value. The resulting trust feedback of the filtered reliable users is aggregated in order to calculate the current trust value of each SP for the specific service. Since more recent trust feedback is more meaningful, a discount factor is used to weigh up a sequence of trust feedback values submitted by a user. Figure 2 illustrates the core processes of trust decision-making in the trust management mechanism. The detailed implementation of these processes is explained in the following sections.

4 The Filtering Mechanism
There are two kinds of unreliable trust feedback, i.e., unfairly positive ratings (ballot stuffing), and unfairly negative ratings (bad mouth) (Whitby et al., 2005). Several methods have been proposed to filter the trust feedback, such as controlled anonymity and cluster filtering (Dellarocas, 2000). These methods are classified as endogenous or exogenous. Endogenous methods are based on analyzing and comparing the rating values in order to filter out unreliable trust feedback. Exogenous methods, on the other hand, use external factors to determine a weight that is applied to all the ratings. The controlled anonymity method is an endogenous method, whereas the cluster filtering is an exogenous method. The method described in this section falls in the integration of endogenous and exogenous methods.

For each assessor we calculate a reliability value based on two factors: familiarity of the assessors with the service provider, and consistency of the assessors’ ratings. Below, we describe these two factors.

The familiarity of an assessor with a specific service offered by a service provider is represented by a) the frequency of usage, b) the duration of the time during which the assessor has been using the service, and c) the last time the assessor used the service. Obviously, the longer and more frequently an assessor uses a service, the more familiar he is with it. However, the length and frequency are not as important if the assessor has interacted with the service sometime in the past. For example, if a user used a service a long time ago and never used it again, then his assessment are not very reliable, if we assume that the QoS of the service is not static. Based on these considerations, we have obtained the following metric for the familiarity.

Let $ASP$ be the set of $N$ alternative service providers who can deliver a particular service that satisfies the functional requirements of a user. We have $ASP = \{ASP_i, ASP_2, ..., ASP_N\}$, where $ASP_i$ indicates the $i$th service provider. Let $M_i$ be the number of users that have used the specific service offered by $ASP_i$, $i=1,2,...,N$. Also, let $X_{ij}(k), k=1,2,...,$ $m_{ij}$ be the be $k$th trust value submitted by the $j$th assessor for the $i$th service provider, $i=1,2,...,N$, and $j=1,2,..,M_i$, with $m_{ij}$ being the total number of trust values. Let $X_{i}(t_0)$ and $X_{i}(t_f)$ be the first and last time the $j$th assessor used the $i$th service provider. Finally, let $T_{c}$ be the current time, and $l_i$ be the length of the time that $ASP_i$ has been offering the specific service. Then, the percent of time $l_i$ that user $j$ has been using the $i$th service is

$$l_i = \frac{X_{i}(t_f) - X_{i}(t_0)}{l_i}$$  \hspace{1cm} (1)

and the familiarity $f_{ij}$ of user $j$ for the specific service offered by the $i$th service provider is obtained as follows:

$$f_{ij} = \left(1 - e^{-\frac{m_{ij}}{\delta}}\right) \delta $$  \hspace{1cm} (2)

where $\delta$ is a discounting factor between (0,1), which indicates that the closer the last interaction to the current time is, the higher the familiarity. We note that familiarity takes values in $[0, 1]$. As the number of interactions $m_{ij}$ and the percent of time $l_i$ that user $j$ has been using the $i$th service grow, the familiarity also increases infinitely to 1. Besides, the $\delta$ can discount the familiarity when comparing the time of last interaction and current time.

The consistency of an assessor indicates whether the feedback given by the assessor for a specific service offered by a service provider is consistent with the overall level of ratings. Let $\mu_i$ be the average of all the ratings submitted by assessor $j$ for the service offered by the $i$th service provider, $\mu_i$ be the average of all the ratings submitted by all assessors for the service offered by the $i$th service provider, and $\mu$ be the average of all ratings submitted by all assessors for the specific service provided by all $N$ service providers. Then, the consistency factor of assessor $j$ for the service provided by the $i$th service provider is calculated as follows:

$$c_{ij} = \left[\frac{\mu_j - \mu_i}{\mu_{max} - \mu_i}\right] \frac{\mu_{min} - \mu_j}{\mu_{min} - \mu_{max}}$$  \hspace{1cm} (3)

where $\mu_{j}$ and $\mu_{i}$ indicate the maximum and minimum value of the difference between the average ratings of the $i$th service provider from user $j$ calculated over all assessors. The consistency of an assessor takes value in the range of $[0, 1]$.

The reliability $r_{ij}$ of assessor $j$ for the rating submitted for the specific service provided by the $i$th service provider is obtained by combining the familiarity $f_{ij}$ with the consistency $c_{ij}$ as follows:

$$r_{ij} = f_{ij} \cdot c_{ij}$$  \hspace{1cm} (4)

Using the reliability factor, the assessors whose reliability is below a threshold are eliminated from calculating the ratings of the service providers. The threshold is specified by a new user who is considering choosing one of the service providers. In addition, for the calculation of the ratings, we also weigh more recent ratings submitted by an assessor using a discount factor $\alpha$. This is because we assume that the trust level of recent ratings is higher than that of earlier ones. Hence, the average $\mu_{ij}$ of the ratings submitted by assessor $j$ for the service offered by the $i$th service provider, is recomputed as follows

$$\mu_{ij} = \frac{\sum_{k=1}^{m_{ij}} \alpha^{k-1} X_{ij}(k)}{\sum_{k=1}^{m_{ij}} \alpha^{k-1}}$$  \hspace{1cm} (5)

We recall that $X_{ij}(k), k=1,2,...,m_{ij}$, be the be $k$th trust
value submitted by the $j$th assessor for the $i$th service provider, $i=1,2,..,N$, and $j=1,2,..,M_i$ with $m_{ij}$ being the total number of the trust values.

The predicted evaluation values from the trust list of target user $t$ can be derived as follows.

$$\hat{\mu}_i = \frac{\sum_{u_j \in U^T} r_{ij} \cdot \mu_{ij}}{\sum_{u_j \in U^T} r_{ij}}$$

(6)

Where $u_j \in U^T$ represents the set of assessors that are reliable to a new user who is considering using the service. This is obtained by eliminating those users whose reliability is below the user-specified threshold.

5 Numerical Results

We carried out simulation experiments in order to establish how well our proposed method filters out biased results. We assume that a cloud service is offered by four different service providers. In addition, we assumed that: (1) service providers do not provide recommendations to users and do not interact with each other; (2) After each transaction, a service user provides his trust feedback to the trust management system, (3) all trust feedback from service users is complete without unknown information.

<table>
<thead>
<tr>
<th>Experiment design</th>
<th>Malicious attack models</th>
<th>Weight of consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>With filtering mechanism</td>
<td>+1, -1, 0</td>
<td>1, 0.9, 0.8, 0.7, ..., 0</td>
</tr>
<tr>
<td>Without filtering</td>
<td>+1, -1, 0</td>
<td>1, 0.9, 0.8, 0.7, ..., 0</td>
</tr>
<tr>
<td>Threshold of reliability</td>
<td>+1, -1, 0</td>
<td>1, 0.9, 0.8, 0.7, ..., 0</td>
</tr>
</tbody>
</table>

We evaluate the trust results’ effectiveness of our reliability model by comparing the trust results using the reliability model and without using the reliability model, and examine the impact of different threshold values of reliability on the trust results. Besides, we will also measure the impact of the relative weights of the consistency and familiarity in our reliability model. The threshold value is a real number in $[0, 1]$. For simplicity, we set the threshold value of users’ reliability 0.6 and 0.7. In order to check the performance of our proposed method, we compare the evaluation results with the situations where all users ratings are taken into account and all the untrusted users are filtered out. We assume the known intrinsic quality of service equal to 3, 4, and 5 respectively.

We assumed that each service provider has an intrinsic quality of service, which is fixed as an input to the simulation. This was varied in order to observe the effectiveness of our proposed filtering method.

Size of participation nodes $N$: Total number of nodes in the simulated cloud environment. We will set different size of participation nodes to evaluate the cloud service quality.

Percentage of dishonest users: The number of service users that provide unreliable feedback assessments over the total number of service users, and these assessments are composed of unfairly positive and unfair negative ratings.

Length of simulation $l$: since the length of time period of interactions impacts the reliability of recommendations, so the performance of our proposed mechanism will be influenced too. We will set different lengths of time period to investigate how it will influence the performance indicators. CI’s are not given in the plots because they are very small and they are not discernible.

Malicious attack models: we consider the following attack models: only with unfairly positive feedback (+1), only with unfairly negative feedback (-1), and with both unfairly positive and negative feedback (0).

We evaluate the trust results’ effectiveness of our reliability model by comparing the trust results using the reliability model and without using the reliability model, and examine the impact of different threshold values of reliability on the trust results. Besides, we will also measure the impact of the relative weights of the consistency and familiarity in our reliability model. The threshold value is a real number in $[0, 1]$. For simplicity, we set the threshold value of users’ reliability 0.6 and 0.7. In order to check the performance of our proposed method, we compare the evaluation results with the situations where all users ratings are taken into account and all the untrusted users are filtered out. We assume the known intrinsic quality of service equal to 3, 4, and 5 respectively.

The reliable users’ trust feedback is in the range of [2, 4], [3, 5], and [4, 5] respectively, the unfairly positive users’ trust feedback is in the range of [3, 4] and [4, 5], and there will be no unfairly positive users when the intrinsic trustworthiness of service is equal to 5. On the other hand, the unfairly negative users’ ratings are in the range of [1, 2], [1, 3], and [1, 4], respectively. Figures 3 to 9 demonstrate the evaluation results under different experiment settings. In these figures, the x-axis indicates the weight of the consistency, and the y-axis means the evaluated trust result.

The following Figures 3-5 show the service trust level evaluation results of service 1, which is in the intrinsic trust level of 3.
The above figures show the trust evaluation results of four different cloud services. It can be clearly inferred that the accuracy of predicted trust decision results are higher than the bottom line of using all users’ feedback without filtering, no matter under which experiment settings, i.e., the trust results with our reliability model are more closer to the intrinsic trustworthiness of cloud services. Moreover, it is clearly that the trust results derived by only considering the consistency factor are more better than those which are derived by only considering the familiarity, especially when the unfairly users rate is low (e.g., when the unfairly use rate=2000/20000=0.1). This is because there are not many unreliable users who return their unfair trust feedback to manipulate the trust results.

6 Conclusions

This paper introduces an effective and robust reliability-based trust management system for cloud environments. The trust management system is able to bridge the trust gap between cloud service providers and users. Users can send back their trust feedback to the trust management system, so that the users who have no experience with the service provider can make decisions according to the trust management mechanism. The feedback from users is filtered according to their reliability, which is depending on two important factors, i.e., the familiarity and the consistency. Furthermore, the trust management system applies the reliability weighted trust value as the basis of trust decisions for users. In the end, the experiment results show that our mechanism and trust model are effective.
References

Shen Zhidong, Li Li, Yan Fei and Wu Xiaoping. Cloud Computing System Based on Trusted Computing Platform. Proceedings of International Conference on Intelligent Computation Technology and Automation
Zheng Zibin, Ma Hao, Micheal R. Lyu, and Irwin King. WSRec: A collaborative filtering based web service recommender system. 2009 IEEE International Conference on Web Services, 437-444