Trust as a Service: A Framework for Trust Management in Cloud Environments

Talal H. Noor and Quan Z. Sheng

School of Computer Science,
The University of Adelaide, Adelaide SA 5005, Australia
{talal,qsheng}@cs.adelaide.edu.au

Abstract. Trust is one of the most concerned obstacles for the adoption and growth of cloud computing. Although several solutions have been proposed recently in managing trust feedbacks in cloud environments, how to determine the credibility of trust feedbacks is mostly neglected. In addition, managing trust feedbacks in cloud environments is a difficult problem due to unpredictable number of cloud service consumers and highly dynamic nature of cloud environments. In this paper, we propose the "Trust as a Service" (TaaS) framework to improve ways on trust management in cloud environments. In particular, we introduce an adaptive credibility model that distinguishes between credible trust feedbacks and malicious feedbacks by considering cloud service consumers' capability and majority consensus of their feedbacks. The approaches have been validated by the prototype system and experimental results.

Keywords: Trust Management, Cloud Computing, Distributed Computing, Credibility Model.

1 Introduction

Over the past few years, cloud computing is gaining a considerable momentum as a new computing paradigm for providing flexible services, platforms, and infrastructures on demand [1,3]. For instance, it only took 24 hours, at the cost of merely $240, for the New York Times to archive its 11 million articles (1851-1980) using Amazon Web Services[1].

Given the quick adoption of cloud computing in the industry, there is a significant challenge in managing trust among cloud service providers and cloud service consumers [1,3,8]. Recently, the significance of trust management is highly recognized and several solutions are proposed to assess and manage trust feedbacks collected from participants [8,5]. However, one particular problem has been mostly neglected: to what extent can these trust feedbacks be credible. Trust management systems usually experience malicious behaviors from its users. On the other hand, the quality of trust feedbacks differs from one person to another, depending on how experienced she is. This paper focuses on the cloud service

[1] http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/

A. Bouguettaya, M. Hauswirth, and L. Liu (Eds.): WISE 2011, LNCS 6997, pp. 314–321, 2011.

consumers perspective (i.e., cloud service consumers assess the trust of cloud services). In particular, we distinguish several key issues of the trust management in cloud environments including i) *Trust Results Accuracy*: determining the credibility of trust feedbacks is a significant challenge due to the overlapping interactions between cloud service consumers and cloud service providers. It is difficult to know how experienced a cloud consumer is and from whom malicious trust feedbacks are expected that requires extensive probabilistic computations [17,9]; ii) *Trust Feedback Assessment and Storage*: the trust assessment of a service in existing techniques is usually centralized, whereas the trust feedbacks come from distributed trust participants. Trust models that use centralized architectures are prone to scalability and security issues [7].

In this paper, we overview the design and implementation of the *Trust as a Service* (TaaS) framework. This framework helps distinguish between the credible and the malicious trust feedbacks through a credibility model. In a nutshell, the salient features of the TaaS framework are i) *A Credibility Model*: we develop a credibility model that not only distinguishes between trust feedbacks from experienced cloud service consumers and from amateur cloud service consumers, but also considers the *majority consensus* of feedbacks; ii) *Distributed Trust Feedback Assessment and Storage*: to avoid the drawbacks of centralized architectures, our trust management service allows trust feedback assessment and storage to be managed distributively.

The remainder of the paper is organized as follows. The design of the TaaS framework is presented in Section 2. Details of the trust management service (TMS) including the distributed trust feedback collection and assessment are described. Section 3 describes the credibility model. Section 4 reports the implementation and several experimental evaluations. Finally, Section 5 discusses the related work and provides some concluding remarks.

2 The Framework

We propose a framework using the Service Oriented Architecture (SOA) to deliver trust as a service. SOA and Web services are one of the most important enabling technologies for cloud computing in the sense that resources (e.g., software, infrastructures, and platforms) are exposed in clouds as services [6,16]. In particular, our framework uses Web services to span several distributed TMS nodes that expose interfaces so that trust participants (i.e., the cloud service consumers) can give their trust feedbacks or inquire about the trust results based on SOAP or REST [15] messages. Figure 1 depicts the framework, which consists of three different layers, namely the *Cloud Service Provider Layer*, the *Trust Management Service Layer*, and the *Cloud Service Consumer Layer*.

– The Cloud Service Provider Layer. This layer consists of different cloud service providers who provide cloud services. The minimum indicative feature that every cloud service provider should have is to provide the infrastructure as a service (i.e., the cloud provider should have a data center that provides the storage, the process, and the communication).

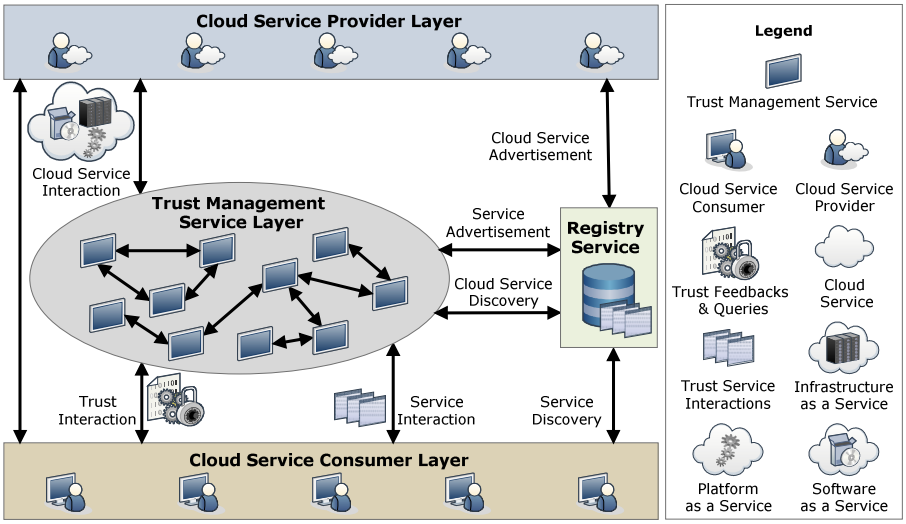


Fig. 1. Architecture of the Trust as a Service Framework

- The Trust Management Service Layer. This layer consists of several distributed TMS nodes that expose interfaces so that cloud service consumers can give their trust feedbacks or inquire about the trust results represents.
- The Cloud Service Consumer Layer. Finally, this layer consists of different cloud service consumers who consume cloud services. For example, a new startup that has limited funding can consume cloud services (e.g., hosting their services in Amazon S3). A cloud service consumer can give trust feedbacks of a particular cloud service by invoking the TMS (see Section 2.1).

Our framework also contains a *Registry Service* (see Figure 1) that has several responsibilities including i) *Service Advertisement*: both cloud service providers and the TMS are able to advertise their services through the *Service Registry*; ii) *Service Discovery*: the TMS and cloud service consumers are able to access the *Service Registry* to discover services.

2.1 Trust Feedback Collection and Assessment

In our framework, the cloud service trust behavior is represented by a collection of invocation history records denoted as $\mathcal{H}$. Each cloud service consumer c holds her point of view regarding the trustworthiness of a specific cloud service s which is managed by the assigned TMS. $\mathcal{H}$ is represented in a tuple that consists of the cloud consumer primary identity $\mathcal{C}$, the cloud service identity $\mathcal{S}$, a set of trust feedbacks $\mathcal{F}$ and the aggregated trust feedbacks weighted by the credibility $\mathcal{F}_c$,

i.e., $\mathcal{H} = (\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{F}_c)$. Each trust feedback in $\mathcal{F}$ is represented in numerical form in which the range of the normalized feedback is [0, 1], where 0, +1, and 0.5 means negative, positive, and neutral respectively. Whenever a cloud consumer inquires the TMS about the trustworthiness of a cloud service s, the trust result ($\mathcal{T}r(s)$), is calculated using:

$$\mathcal{T}r(s) = \frac{\sum_{l=1}^{|\mathcal{V}(s)|} \mathcal{F}_c(l,s)}{|\mathcal{V}(s)|} \tag{1}$$

where $\mathcal{V}(s)$ is all trust feedbacks given to the cloud service s and $|\mathcal{V}(s)|$ represents the length of the $\mathcal{V}(s)$. $\mathcal{F}_c(l,s)$ are trust feedbacks from the l^{th} cloud consumer weighted by the credibility.

The TMS distinguishes between credible trust feedbacks and malicious trust feedbacks through assigning the *Cloud Consumer's Experience* aggregated weights $Exp(l)$ to trust feedbacks $\mathcal{F}(l,s)$ as shown in Equation 2, where the result $\mathcal{F}_c(l,s)$ is held in the invocation history record h and updated in the assigned TMS.

$$\mathcal{F}_c(l,s) = \mathcal{F}(l,s) * Exp(l) \tag{2}$$

3 Credibility Model

There is a considerable possibility that the TMS receives *inaccurate* or even *malicious* trust feedbacks from amateur cloud service consumers (e.g., who lack experience) or vicious cloud service consumers (e.g., who submit lots of negative feedbacks to disadvantage a particular cloud service). To overcome these issues, we propose a *credibility model*, which is centered on the *cloud consumer's experience*. To differentiate between expert and amateur cloud service consumers, we consider the *Majority Consensus* and the *Cloud Consumer's Capability*.

Majority Consensus. It is well-known that the majority of people usually agree with experts' judgments about what is good [4]. Similarly, we believe that the majority of cloud consumers agree with *Expert* cloud service consumers' judgments. In other words, any cloud service consumer whose trust feedback is close to the majority of trust feedbacks is considered an *Expert Cloud Service Consumer* (ECSC), or an *Amateur Cloud Service Consumer* (ACSC) otherwise. In order to measure how close the cloud service consumer's trust feedbacks to the majority (i.e., the *Majority Consensus* ($\mathcal{J}(c)$) which is calculated as follows:

$$\mathcal{J}(c) = 1 - \sqrt{\frac{\sum_{h \in \mathcal{V}c(c)} \left(\sum_{k=1}^{|\mathcal{V}c(c,k)|} \left(\frac{\mathcal{F}(c,k)}{|\mathcal{V}c(c,k)|} - \left(\frac{\sum_{l \neq c, l=1}^{|\mathcal{V}c(l,k)|} \mathcal{F}(l,k)}{|\mathcal{V}(k)| - |\mathcal{V}c(c,k)|} \right) \right) \right)^2}{|\mathcal{V}c(c)|}} \tag{3}$$

where the first part of the numerator represents the mean of the cloud service consumer c's trust feedbacks $\mathcal{F}(c,k)$ for the k^{th} cloud service. The second part of

the numerator represents the mean of the majority trust feedbacks given by other cloud service consumers ($\mathcal{F}(l,k)$) (i.e., the l^{th} cloud service consumer, except the cloud service consumer c) to the k^{th} cloud service.

Cloud Service Consumer's Capability. It is a common sense that older people are likely to be more experienced in judging things than younger people [14]. However, this is only true if the older people have experienced considerable number of judging practices. As a result, we believe that "older" cloud service consumers who have many judging practices are likely to be more experienced and capable. A cloud service consumer's capability ($\mathcal{B}$) is measured as follows:

$$\mathcal{B}(c) = \begin{cases} 1 + \frac{|\mathcal{V}c(c)|}{\mathcal{A}g(c)} & if \quad |\mathcal{V}c(c)| \leq \mathcal{A}g(c) \\ 2 & otherwise \end{cases} \tag{4}$$

where $\mathcal{V}c(c)$ represents all good feedbacks (i.e., feedbacks which are close to the majority) given by the cloud service consumer c. $\mathcal{A}g(c)$ denotes the virtual *Age* of a certain cloud service consumer, measured in days since the registration in the TMS. The idea behind adding the number 1 to this ratio is to increase the value of a cloud service consumer experience based on $\mathcal{B}(c)$ result. In other words, we use $\mathcal{B}(c)$ as a *reward* factor. The higher $\mathcal{B}(c)$ is, the more experienced a cloud service consumer is. It should be noted that even if a malicious cloud service consumer attempts to manipulate the capability result, the capability result will not exceed 2.

Based on the specified cloud service consumer's experience factors (i.e., $\mathcal{B}(c)$ and $\mathcal{J}(c)$), the TMS distinguishes between ECSC and ACSC through assigning the cloud service consumer's *Experience* aggregated weights $Exp(c)$ to each of the cloud consumers' trust feedbacks as shown in Equation 2. $Exp(c)$ is calculated as follows:

$$Exp(c) = \frac{\beta * \mathcal{B}(c) + \mu * \mathcal{J}(c)}{\lambda} \tag{5}$$

where β and $\mathcal{B}(c)$ denote the *cloud service consumer's Capability* factor's normalized weight and the factor's value respectively. The second part of the equation represents the *Majority Consensus* factor where μ denotes the factor's normalized weight and $\mathcal{J}(c)$ denotes the factor's value. λ represents the number of factors used to calculate $Exp(c)$ (e.g., if we only consider cloud service consumer's capability, $\lambda = 1$; if we consider both cloud service consumer's capability and majority consensus, $\lambda = 2$).

We use $\mathcal{J}(c)$ as a *penalty* factor (i.e., because $\mathcal{J}(c)$ ranges [0,1] as described in equation 3). The lower $\mathcal{J}(c)$ is, the lower the experience of the cloud service consumer c is. However, $\mathcal{B}(c)$ is used as a reward factor (i.e., because $\mathcal{B}(c)$ ranges [1, 2] as described in equation 4). Higher $\mathcal{B}(c)$ means more experienced of a cloud service consumer. It is worth mentioning that our credibility is dynamic and is able to detect behavior changes. For example, if a cloud service consumer behaves good for a period of time (e.g., to gain credibility) and then starts misbehaving, $\mathcal{J}(c)$ can detect such behavior through applying the standard deviation.

4 Implementation and Experimental Evaluation

Our implementation and experiments were developed based on the NetLogo platform[2], which was used to simulate the cloud environments. We particularly focused on validating and studying the performance of the proposed credibility model (see Section 3). In our experiments, we used real-life trust data set, Epinions[3] rating data set which was collected by Massa and Avesani [13]. We choose to use Epinions data set because its data structure is similar (i.e., consumers opinions and reviews on specific products and services) to our cloud service consumer trust feedbacks. The data set has 49,290 users, 139,738 items, and 664,824 trust feedbacks.

Table 1. Experiment Factors and Parameters Setup

Experiment Design	β	μ	λ	$Exp(c)$
With Credibility Factors	1	1	2	
Without Credibility Factors				1
Cloud Service Consumer's Capability Factor	1	0	1	
Majority Consensus Factor	0	1	1	

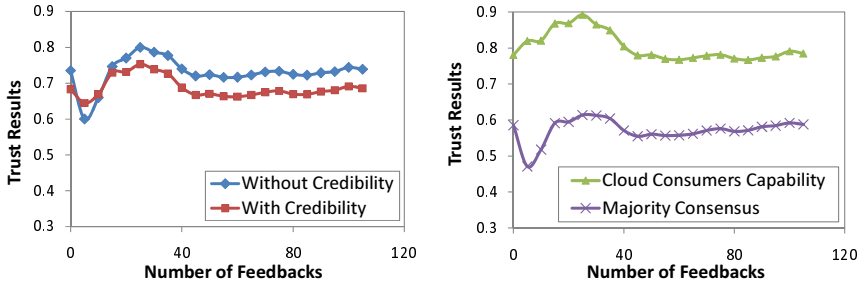
We evaluate our credibility model using both *analytical analysis* and *empirical analysis*. The analytical analysis focuses on measuring the trust result accuracy when using the credibility model and without using the credibility model (i.e., we turn the $Exp(c)$ to 1 to exclude the credibility factor). The empirical analysis focuses on measuring the trust result accuracy for each factor in our credibility model (i.e., $\mathcal{B}(c)$ and $\mathcal{J}(c)$). The parameters setup for each corresponding experiment are depicted in Table 1.

Figure 2(a) depicts the analytical analysis of the trust results for a particular cloud service. We note that the trust results are oscillating more significantly when calculating the trust without considering the credibility factors than when calculating the trust with credibility factors. In other words, even if the TMS receives inaccurate or malicious trust feedbacks, it is difficult to manipulate the trust results by using our credibility model.

Figure 2(b) shows the empirical analysis of the same cloud service. We note that trust results obtained by only considering $\mathcal{B}(c)$ are higher than the trust results by only considering $\mathcal{J}(c)$. This is true, because we use $\mathcal{B}(c)$ as a reward factor and the $\mathcal{J}(c)$ as a penalty factor. This reflects how adaptive our credibility model is where the credibility factors can easily be tweaked according to the TMS's needs. For instance, for optimistic situations where only a few cloud service consumers have high values of capability, increasing the cloud service consumer's capability factor (i.e., β) will help the TMS to distinguish between experienced cloud consumers and inexperienced ones. On the other hand, for pessimistic situations where many cloud consumers have high values of capability, the majority consensus factor (i.e., μ) needs to be increased.

[2] http://ccl.northwestern.edu/netlogo/

[3] http://www.trustlet.org/wiki/Downloaded_Epinions_dataset



(a) Analytical Analysis (b) Empirical Analysis

Fig. 2. Experimental Evaluation

5 Discussions and Conclusion

Trust management is one of the critical issues in cloud computing and a very active research area [10,12,8,2]. For instance, Hwang et al. [8] proposed a security-aware cloud architecture where trust negotiation and data coloring techniques are used to support the cloud service provider perspective. The cloud service consumer's perspective is supported using the trust-overlay networks to deploy a reputation-based trust management. Brandic et al. [2] proposed a centralized approach for compliance management in cloud environments that supports the cloud service consumer's perspective using compliant management to help the cloud service consumers in selecting proper cloud services. Unlike previous works that use centralized architecture, we present a credibility model supporting distributed trust feedback assessment and storage. This credibility model also distinguishes between trustworthy and malicious trust feedbacks.

Conner et al. [5] proposed a decentralized trust management framework for SOA that supports the service provider's perspective. This framework offers multiple trust evaluation metrics to allow customized evaluation and assessment of service consumers. Malik and Bouguettaya [11] proposed decentralized reputation assessment techniques based on the existing quality of service (QoS) parameters. The proposed framework supports different assessment metrics such as rater credibility, past rating history, etc. Unlike previous works that require trust participants' collaboration by rating trust feedbacks, our credibility model distinguishes between trustworthy and malicious trust feedbacks without such technique. We were inspired by Xiong and Liu who differentiate between the credibility of a peer and the credibility of the feedback [18]. However, this approach is inappropriate in cloud environments because peers give and receive services and they are evaluated on that base. In other words trust results are used to distinguish between credible and malicious feedbacks.

In this paper, we have presented a "Trust as a Service" framework to manage trust in cloud environments. We introduced an adaptive credibility model that assesses cloud services' trustworthiness and distinguishes between credible and malicious trust feedbacks. We particularly introduced the cloud service consumer's *Capability* and the *Majority Consensus* factors in calculating the

trust of a cloud service. In addition, our TMS allows trust feedback assessment and storage to be managed in a distributed way. In the future, we plan to deal with more challenging problems such as the *Sybil* attack and the *Whitewashing* attack. Performance optimization of TMS is another focused work.

References

1. Armbrust, M., et al.: A View of Cloud Computing. Communiaction of the ACM 53(4), 50–58 (2010)
2. Brandic, I., Dustdar, S., Anstett, T., Schumm, D., Leymann, F., Konrad, R.: Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. In: Proc. of IEEE CLOUD 2010, Miami, Florida, USA (July 2010)
3. Buyya, R., Yeo, C., Venugopal, S.: Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities. In: Proc. of IEEE HPCC 2008, Dalian, China (September 2008)
4. Child, I.: The Psychological Meaning of Aesthetic Judgments. Visual Arts Research 9(2(18)), 51–59 (1983)
5. Conner, W., Iyengar, A., Mikalsen, T., Rouvellou, I., Nahrstedt, K.: A Trust Management Framework for Service-Oriented Environments. In: Proc. of WWW 2009, Madrid, Spain (April 2009)
6. Dillon, T., Wu, C., Chang, E.: Cloud Computing: Issues and Challenges. In: Proc. of AINA 2010, Perth, Australia (April 2010)
7. Hoffman, K., Zage, D., Nita-Rotaru, C.: A Survey of Attack and Defense Techniques for Reputation Systems. ACM Computing Surveys 42(1), 1–31 (2009)
8. Hwang, K., Li, D.: Trusted Cloud Computing with Secure Resources and Data Coloring. IEEE Internet Computing 14(5), 14–22 (2010)
9. Jøsang, A., Quattrociocchi, W.: Advanced Features in Bayesian Reputation Systems. In: Fischer-Hübner, S., Lambrinoudakis, C., Pernul, G. (eds.) TrustBus 2009. LNCS, vol. 5695, pp. 105–114. Springer, Heidelberg (2009)
10. Krautheim, F., Phatak, D., Sherman, A.: Introducing the Trusted Virtual Environment Module: A New Mechanism for Rooting Trust in Cloud Computing. In: Acquisti, A., Smith, S.W., Sadeghi, A.-R. (eds.) TRUST 2010. LNCS, vol. 6101, pp. 211–227. Springer, Heidelberg (2010)
11. Malik, Z., Bouguettaya, A.: RATEWeb: Reputation Assessment for Trust Establishment Among Web services. The VLDB Journal 18(4), 885–911 (2009)
12. Manuel, P., Thamarai Selvi, S., Barr, M.E.: Trust Management System for Grid and Cloud Resources. In: Proc. of ICAC 2009, Chennai, India (December 2009)
13. Massa, P., Avesani, P.: Trust Metrics in Recommender Systems. In: Computing with Social Trust. Human-Computer Interaction Series. Springer, Heidelberg (2009)
14. Roosevelt, E.: Facing the problems of youth. The P.T.A. magazine: National Parent-Teacher Magazine 29(30), 1–6 (1935)
15. Sheth, A.P., Gomadam, K., Lathem, J.: SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. IEEE Internet Computing 11(6), 84–87 (2007)
16. Wei, Y., Blake, M.B.: Service-oriented Computing and Cloud Computing: Challenges and Opportunities. IEEE Internet Computing 14(6), 72–75 (2010)
17. Weng, J., Miao, C., Goh, A.: Protecting Online Rating Systems from Unfair Ratings. In: Katsikas, S.K., López, J., Pernul, G. (eds.) TrustBus 2005. LNCS, vol. 3592, pp. 50–59. Springer, Heidelberg (2005)
18. Xiong, L., Liu, L.: Peertrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities. IEEE TKDE 16(7), 843–857 (2004)