Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 617087, 9 pages
http://dx.doi.org/10.1155/2014/617087

Research Article

Reputation Revision Method for Selecting Cloud Services Based on Prior Knowledge and a Market Mechanism

Qingtao Wu, Xulong Zhang, Mingchuan Zhang, Ying Lou, Ruijuan Zheng, and Wangyang Wei

Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Qingtao Wu; wuqingtao.cn@hotmail.com

Received 12 October 2013; Accepted 4 November 2013; Published 17 February 2014

Academic Editors: W. Sun, G. Zhang, and J. Zhou

The trust levels of cloud services should be evaluated to ensure their reliability. The effectiveness of these evaluations has major effects on user satisfaction, which is increasingly important. However, it is difficult to provide objective evaluations in open and dynamic environments because of the possibilities of malicious evaluations, individual preferences, and intentional praise. In this study, we propose a novel unfair rating filtering method for a reputation revision system. This method uses prior knowledge as the basis of similarity when calculating the average rating, which facilitates the recognition and filtering of unfair ratings. In addition, the overall performance is increased by a market mechanism that allows users and service providers to adjust their choice of services and service configuration in a timely manner. The experimental results showed that this method filtered unfair ratings in an effective manner, which greatly improved the precision of the reputation revision system.

1. Introduction

The rapid developments of cloud computing means that cloud services have become the main computing mode on the Internet. Many services have been deployed to provide similar functionalities. However, the problem of identifying reliable services has attracted the attention of researchers [1, 2]. Thus, the concepts of trust and reputation [3] have been introduced to assess the reliability of cloud services.

Reputation is a subjective assessment of a cloud service, which is based on individual experience or the recommendations of other users. Reputation and trust are dynamic, which makes the construction of an evaluation standard a challenging task. In addition, the occurrence of malicious evaluations [4, 5], deliberate praise, and the personal preferences of users means that a standardized reputation value may differ from the true value.

Recently, various reputation revision systems have been proposed to address the challenges posed by open and dynamic cloud service environments [6]. Most of these systems are focused on the calculation of reputation ratings, reputation management, experience, and other features of dynamic environments [7–10] that might provide an appropriate reference for users. However, the existence of unfair ratings greatly affects the accuracy of trust evaluations. Currently, these reputation models are mainly tending to the accuracy of trust evaluations [11, 12]; however, these existing methods are limited by personality preference.

Based on historical user evaluations and preferences related to specific requirements, we propose a method that revises the reputation rating. This prior knowledge is combined with a filtering algorithm based on the similarities of evaluations, which can distinguish between unfair ratings in an effective manner. This algorithm also includes a market mechanism that allows users and services to act as buyers and sellers. Our method uses only the user feedback ratings data related to a service. Extensive experiments showed that our method distinguished and filtered unfair ratings correctly, while it could also recommend appropriate services for a specific user based on their preferences in a dynamic market environment.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of related work on reputation revision methods. Section 3 describes the problem and provides definitions related to a specific scenario. Section 4 explains our proposed reputation revision framework and Section 5 presents the experimental results. In Section 6, we conclude with the discussion and we summarize the contributions of this study.

2. Related Work

Reputation is a term that has different meanings in various domains. In our study, reputation is defined as an indicator of whether a user is willing to select a service based on the evaluations of other users. Thus, the result of a reputation evaluation will affect the decision about whether to interact with a service provider. Feedback related to previous interactions among users and service providers is collected by a reputation system to predict its future reliability [13, 14]. Reputation evaluations are essential parts of many recommendation systems [15]. Many systems are located in a central server, which can access, collect, and evaluate historical reputation scores from a large number of users [16–19]. Yang et al. [9] proposed a reputation management framework for service selection based on similarity theory. Different weights were set to compute the reputation based on the unique recommendations of users. However, the management framework was designed for a centrally controlled server, which was not suitable for a dynamic cloud service environment.

Several reputation evaluation approaches have been proposed for distributed systems. Ghaffarinejad and Akbari [10] introduced a distributed reputation mechanism, which was based on a number of special reputation centers. Each special reputation center collected reputation information for predetermined services offered by different service providers. However, this method was still somewhat centralized. Faniyi and Bahsoon [7] proposed a decentralized resource control mechanism, which introduced a market-oriented cloud computing architecture. However, the reputation system was vulnerable to whitewashing, incorrectly reported feedback, and collusion attacks (where several users coordinate their feedback to manipulate reputation information).

Kussul et al. [20] focused on the analysis of security threats in trust models and assessed the most important and critical security threats for a utility-based reputation model based on grids. Dong-Sheng et al. [21] proposed a distributed trust mechanism, which calculated two reputation values (for a seller and a buyer) for each node in an iterative manner based on the transaction history. This mechanism could rapidly reduce the reputation values of malicious nodes and prevent collusion attacks. To construct a trustworthy computing environment, Gui et al. [19] proposed a penalty-incentive mechanism based on a repeated game theory, which included a rule related to rewards and punishments. Hawa et al. [16] introduced enhanced reputation-based cooperation incentives, which facilitated better detection and control of free riders. This approach enhanced the scalability and fairness of the system.

Kim and Phalak [18] proposed a computational trust framework for predicting the degree of trust. In addition, Su et al. [12] developed a priority-based trust model, which determined the trustworthiness of a service provider based on designated referees and its historical performance. This method only used third-party evaluations of the previous overall performance of the service and did not consider the individual preferences of the current user. Gorner et al. [22] proposed several improvements for trust modeling, including limiting the size of the advisor network either by specifying the maximum size of a buyer's advisor network or by setting a minimum trustworthiness threshold for agents accepted into the advisor network.

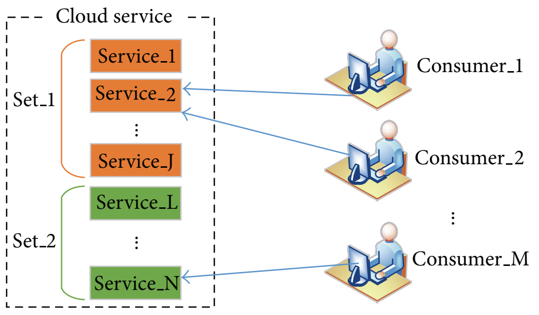

Figure 1: Cloud service selection model.

Another previous study [23] proposed a method for revising reputation values that calculated the reputation based on the difference between the advertised quality of service (QoS) provided by service providers and the evaluations made by consumers. Next, the consumers were sorted based on the reputation ratings they provided and those consumers that might be involved with collusion were mined using an association rules algorithm. Finally, the updated reputation was recalculated and saved in the reputation center.

3. Overview

3.1. Problem Description and Scenario Definition. Our method was developed to overcome some of the limitations of existing reputation mechanisms and is suitable for open and dynamic environments. Before providing the details of our proposed reputation revision mechanism, it is necessary to define the scope of our study and to explain some of the definitions used in this paper.

A cloud computing system provides services according to a third-party mechanism. Thus, users only need to be concerned with the service provided by the cloud. The cloud service selection model shown in Figure 1 contains two agent types, that is, consumer agents and service agents. Services that share the same functionality are placed in the same sets.

Users often select services from the same functional groups based on their own experiences and those of other consumers. However, unreliable evaluations may mean that the reputation of a service does not represent their actual reliability. Some service providers pay consumers to give them high scores for their services or to give low scores to their competitors. Furthermore, it may be difficult to satisfy users with specific requirements. These issues may mean that the service scores are not effective reference sources for users.

Definition 1. A service can be described as a 2-tuple; that is, Service = (Function, QoS). Function is a set of common properties, where different services are classified into separate sets. The main components of QoS include the response time (RT), cost (C), and reliability (R). In the present study, we define QoS as a 3-tuple; that is, QoS = (RT, C, R).

Definition 2. The rating score (RS) is derived from the distributed consumer agents. RT represents the degree of satisfaction with the service. RT is defined as a 3-tuple; that is, RS = (V_RT, V_C, V_R), where V_RT, V_C, and V_R are the values of the properties of the service given in Definition 1. We use percentile scores to distinguish between good and bad performance.

Definition 3. The consumer rating is the major focus, and the consumer is associated with multiple services from different providers. Thus, the consumer agent can be described as follows.

Consumer = (C_ID, $\bigcup_i$ Service_i, $\bigcup_j$ RS_j), where C_ID is the consumer's identifier and $\bigcup_i$ Service_i and $\bigcup_j$ RS_j are the services and rating scores associated with the consumer, respectively.

3.2. Overview of the Entire Reputation System. In this section, we briefly introduce our reputation system structure. The framework of the reputation system is shown in Figure 2.

In general, the framework is applicable to most of the ratings-based experiences shared on online platforms where users evaluate services with numerical ratings. Figure 2 shows that the users ratings are collected as feedback to reputation processing node and the feedback is quantified based on the QoS attributes in the knowledge repository. After normalizing the feedback data, the ratings are filtered based on similarity classification. Next, we set the preferences for abnormal users, who may have specific service requirements. The user preferences mean that the recommendation system provides the most relevant services. There is also a certain degree of punishment for collusive users, who make the results confusing when service recommendations are required. The core of the system is the reputation calculation, where we combine the result from filtering with the historical reputation to generate a reputation value that is credible and reliable. The final part of the reputation framework is reputation management. Using a market mechanism, the service provider can optimize his service configuration and the user can optimize his decision. Thus, the service quality is optimized for the overall environment using the reputation system. Filtering abnormal users also makes the reputation of the service more accurate.

4. Reputation Revision Mechanism

In this section, we describe the mechanism used to ensure a more accurate level of consumer trust in dynamic environments. As mentioned in Section 3, consumers evaluate services using numerical ratings. The proposed method estimates the degree of trust a consumer places in a service based on the consumer's preference and it filters the abnormal reputation ratings.

Table 1: Quality of service parameters for a cloud service.

	p_1	p_2	p_3	p_4
s_1	0.9	0.6	0.7	0.4
s_2	0.8	0.7	0.6	0.3
s_3	0.6	0.6	0.4	0.4
s_4	0.5	0.2	0.2	0.8

Table 2: Reputations of services evaluated by consumers.

	s_1	s_2	s_3	s_4
c_1	0.9	0.6	0.7	0.4
c_2	0.8	0.7	0.6	0.3
c_3	0.6	0.6	0.4	0.4
c_4	0.5	0.2	0.2	0.8

4.1. Filtering the Abnormal Reputation Ratings. The presence of inaccurate assessments affects the overall evaluation of a service to some extent. Thus, we use the similarity to distinguish between abnormal evaluations, which reduces the effects of abnormal reputation ratings. The Euclidean distance is the shortest length of a line in n-dimensional space, which is usually defined as the real distance between two points in n-dimensional space.

We use $S = \{s_1, s_2, \dots, s_n\}$ to represent a service set where services share the same function. $C = \{c_1, c_2, \dots, c_m\}$ is the set of consumers. $P = \{p_1, p_2, \dots, p_s\}$ is the set of QoS parameters [24] for a service. We specify q_{ij} as the parameter values of p_j for s_i, as shown in Table 1. r_{ij} represents the reputation of s_j in c_i, where $0 \le r_{ij} \le 1$, as shown in Table 2.

Each row is regarded as a node. Thus, the similarity between two nodes can be represented by the Euclidean distance. If the distance between two nodes is high, the similarity will obviously be low. Thus, we use the following to compute the similarity between the service parameter and a user's evaluation:

$$\text{similarity} = \frac{1}{\left(\sum_1^n (X_i - Y_i)^2 / n\right)}. \tag{1}$$

In (1), X and Y are vectors, where X_i (or Y_i) represents the value of the i-dimension in the vector. To filter abnormal evaluations, we need to determine whether the parameters of the service configuration have changed. If the parameters of a service have changed, a distance will be generated. In general, there is no change in the configuration of the service parameters. However, if the parameters of the service configuration change, the historical reputation weighting will decline rapidly. If a consumer's evaluation is a considerable distance from the mean of all the other service evaluations made by consumers, we conclude that this consumer's evaluation is problematic and it must be filtered before further processing.

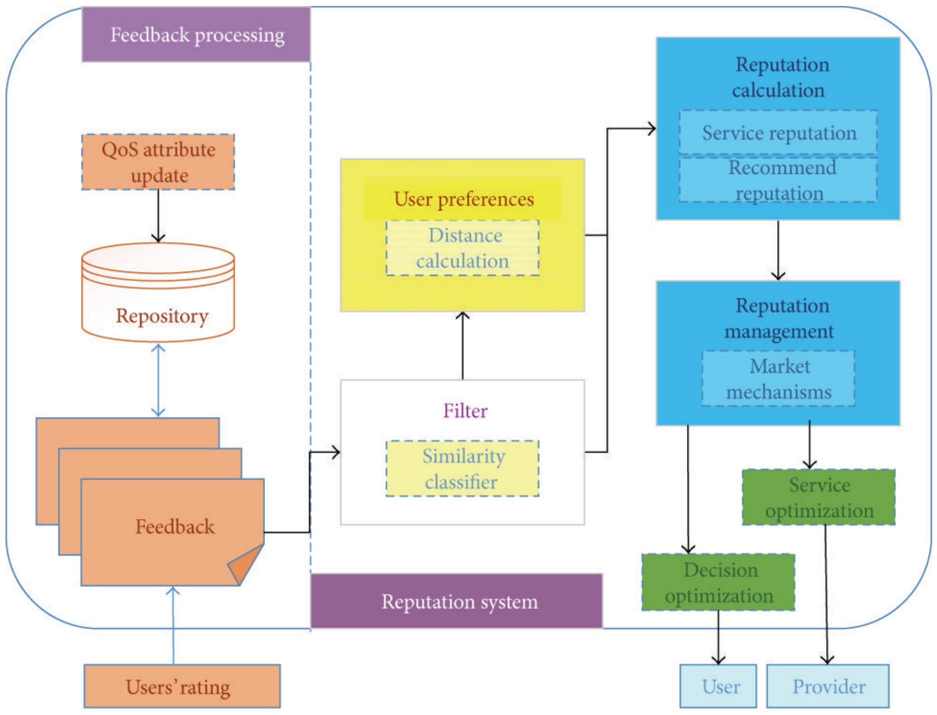

FIGURE 2: Reputation system framework.

4.2. Estimation of Consumer Preference. All consumer evaluations are biased to some extent, so we need to set all of the abnormal consumer preferences. In this case, bias refers to subjective evaluations of objective factors by consumers that deviate from the norm. Consumers who make frequent unfair evaluations of services have extreme preferences. Malicious evaluation refers to the intentional denial of the objective facts, undeserved praise, or unmerited negative feedback. Thus, setting preferences allows us to eliminate abnormal evaluations. However, consumers can change their preferences if they have unusual requirements.

The preference weight (PW) represents a consumer's personal service preferences. A higher PW indicates that the rating for a service differs greatly from that of most users. The PW is constructed using two factors, that is, C Value and Ref. C Value represents a consumer's rating score, while Ref describes the contribution to the PW made by the total consumer scores. PW is defined using the following formula:

$$\text{PW} = C\,\text{Value} - \text{Ref}. \tag{2}$$

C Value is the service's current reputation score given by a consumer. According to (3), Ref is the reference value for the overall score. The consumer rating scores have a normal distribution. Ref is the normal position parameter that describes the location of the central tendency of the normal distribution. rateScore = Ref is the normal to the axis of symmetry, which is completely symmetrical. In a normal distribution, the mean, median, and mode are the same; that is, they are equal to Ref. σ indicates the data distribution with a normal degree of dispersion:

$$f(\text{rateScore}) = \frac{1}{\sqrt{2\pi}\sigma} e^{(\text{rateScore}-\text{Ref})^2/2\sigma^2}, \tag{3}$$

$$\text{Ref} = \frac{1}{N}\sum_{i=1}^{N} \text{rateScore}_i, \tag{4}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\text{rateScore}_i - \text{Ref})^2}. \tag{5}$$

The rateScore distribution of a service is shown in Figure 3 (the data used to produce the figure were based on the evaluation of a service). The PWs are set for overall consumer but they just apply to a small minority of consumers who have obviously deviation to the true quality of service level. According to Figure 3, we simply need to set separate PWs for the reputations that are outside the confidence interval. The confidence interval here refers to the proportion of real evaluation,which can be used as the reliability estimation that is usually defined by a large amount of observation. Obviously, an excessive proportion may mislead the service evaluation while a too little one leads to an unauthentic conclusion.

If σ and Ref are known, the service rateScore follows a normal distribution N (Ref, σ). We define the main interval using (6). α is the width of the main interval and δ is the probability of the evaluation given by most users:

$$P\left\{|\text{rateScore} - \text{Ref}| \le f\left(\frac{\alpha}{2}\right)\right\} = \delta. \tag{6}$$

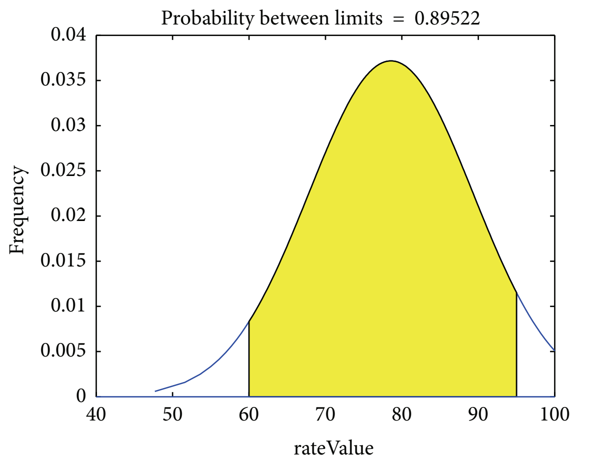

Figure 3: Distribution of a service's rateScore given by consumers.

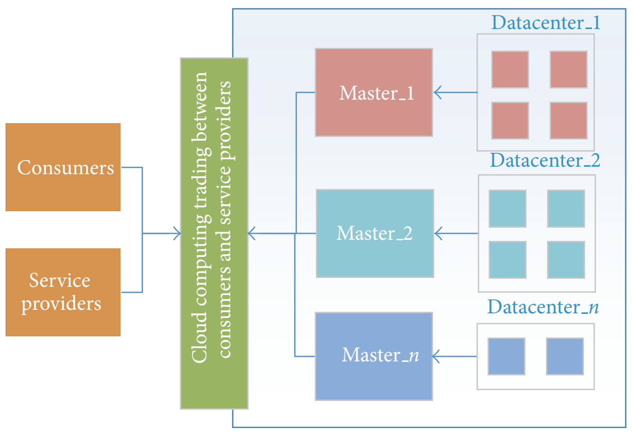

Figure 4: Cloud service market structure.

Thus, we only need to set an appropriate value for δ to distinguish between abnormal evaluations. So we can give the abnormal ratings malicious preference treatment and reduce the weight of them in the whole calculation of service reputation. Suitable recommendations of appropriate services can be made based on the global reputation of a service and the preferences of users, while the system also punishes malicious users.

4.3. Calculation of the Integrated Reputation. The reputation rating of a cloud service given by a specific user is the weighted average of the directly experienced reputation and the historical reputation, which is obtained using (7). RoS_{ij} is the reputation rating based on the direct experience of service j given by consumer i. C_{ij} is the current reputation rating of service j given by consumer i. HsR_{ijk} is the kth reputation rating of the historic reputation rating given by consumer i to service j, and l is the total number of times that consumer i rates service j. β is the weight factor of the current reputation, $0 \le \beta \le 1$

$$\mathrm{RoS}_{ij} = (1-\beta)\, C_{ij} + \frac{\beta}{l}\sum_{k=1}^{l} \mathrm{HsR}_{ijk}. \tag{7}$$

We need to integrate the evaluations of the same service to calculate a service's reputation using the method proposed earlier. After calculating the mean score and the variance (using (8) and (9), resp.) of a service, we can determine its normal distribution (using (10)):

$$\mathrm{Ref}_j = \frac{1}{r}\sum_{i=1}^{r} \mathrm{RoS}_{ij}, \tag{8}$$

$$\sigma^2 = \frac{1}{r}\sum_{i=1}^{r}\left(\mathrm{RoS}_{ij} - \mathrm{Ref}_j\right)^2, \tag{9}$$

$$\mathrm{RoS}_j \sim \left(\mathrm{Ref}_j, \sigma^2\right). \tag{10}$$

Ref_j represents the mean score of the service where the ID is j. r represents the total number of service ratings. RoS_j is the score for j's service. We need to set the value of δ to control the confidence interval. Filtering is more accurate if δ is a low value. However, this increases the complexity of processing the preferences. The specific PW is obtained using the following:

$$\mathrm{PW}_{ij} = \begin{cases} 0 & \left|\mathrm{RoS}_j - \mathrm{Ref}_j\right| \le f\left(\dfrac{\alpha}{2}\right) \\ C_{ij} - \mathrm{Ref}_j & \text{else.} \end{cases} \tag{11}$$

PW_{ij} represents the ith consumer's PW for the jth service. We obtain a more objective service reputation rating after eliminating the personal preferences. The reputation rating of service j is calculated using the following:

$$\mathrm{RoS}_j = \frac{1}{r}\sum_{i=1}^{r}\left(\mathrm{RoS}_{ij} - \mathrm{PW}_{ij}\right). \tag{12}$$

If a consumer wants to access a service, the trust system provides a recommendation based on the reputation using (13). The reputation rating of each consumer is not the same because they have specific preferences, which also facilitates the punishment of malicious evaluations. RR_{ji} represents the reputation rating of service j recommended to consumer i:

$$\mathrm{RR}_{ji} = \mathrm{RoS}_j + \mathrm{PW}_{ij}. \tag{13}$$

4.4. Dynamic Reputation Optimization Using a Market Mechanism. Cloud services are reliant on the dynamic and distributed cloud environment. Thus, a trust system needs to adapt to open and changing conditions. Service providers will change the configurations of their services to meet consumer demands, and the consumers will have variable service preferences. Different consumers use the same service, which runs in different data centers. We use a market mechanism [25] to construct a reputation optimization method that provides a better QoS. The cloud service market environment is shown in Figure 4. The data center, which is a distributed cloud resource provider, includes many nodes. The master node is the seller's agent, which is responsible for changing the service configuration and optimizing the service performance.

The market mechanism includes buyers and sellers. The buyers are consumers and the sellers are service providers in our method. We state the objectives of the buyers and sellers in the cloud market. The buyers' goal is to satisfy their personal demand for a service with a high reputation. The sellers' goal is to receive a better evaluation by optimizing their configuration to maximize the number of tasks completed successfully. The global objective of the market is to supply each buyer with a reliable service while minimizing the costs of the sellers.

The buyers and sellers have different demands, and the cloud service market is open and dynamic. We define the buyer's personal valuation using a utility function, which for buyer b using service i is defined as $P_b(i) = -\alpha\text{Cost}(i) - \beta\text{Time}(i) + \gamma\text{Profit}(i)$, where Cost$(i)$ and Time(i) represent the price and response time for service i, respectively, and Profit(i) is the payoff derived from the usage of service i. The seller's utility function is defined by $P_s(i) = O(i) - C(i)$, where $O(i)$ represents the reputation and fees received for service i and $C(i)$ is the cost of service i. The seller's objective is to maximize $P_s(i)$.

After each transaction, the sellers are rated based on their performance in the task allocated to them. The buyer decides this rating by comparing the actual service completion time with the expected time, as well as with other services. The global objective of the market is to maximize the number of services that are supplied satisfactorily by service providers. This objective is defined using the following:

$$G(i,j) = \text{Max} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(R_{ij} - A_{ij}\right), \quad (14)$$

where R_{ij} is the reputation rating given by buyer i and A_{ij} is the actual performance of service j. Algorithm 1 shows how the market transactions operate between buyers and sellers in pseudocode.

The buyer's timely and correct feedback facilitates the continuous optimization of the service in a dynamic market. If new services and buyers join the market, the system needs to be updated and new items will be available for the buyer to choose. After the service has been updated, a recommendation will be given to the buyer, but the reputation continues to be accumulated.

The market contains many submarkets. Submarkets are synchronized regularly using older market information, which reduces the frequency of management. A service can be present in different submarkets, which are distributed and flexible.

```
(1)  buyer = {$b_1, b_2, \ldots, b_k$}; //the buyer's set
(2)  seller_service = {$s_1, s_2, \ldots, s_l$}; //the service provider's
(3)  services comprise the seller's set.
(4)  maxProfitofBuyer = 0; //buyer pursues their best
     interests
(5)  maxReputationRepay = {0, 0, ..., 0}; //the seller
(6)  receives a high payoff from the buyer, which forms
     a set
(7)  for $i$ = 1 to $k$
(8)      for $j$ = 1 to $l$do
(9)          if $b_i$ has used $s_j$
(10)             RS[$ij$] = the rating provided by $b_i$ for $s_j$;
(11)                                       //collect the
     ratings
(12)         else RS[$ij$] = null;
(13)         end if
(14)     end for
(15) end for
(16) for $j$ = 1 to $l$
(17)     for $i$ = 1 to $k$ do
(18)         RS[$j$] = comput_reputation (RS[$i$, $j$]);
(19)                               //compute service $j$'s
     reputation
(20)     end for
(21) end for
(22) if same_function(service_set) //service_set includes
(23)                   //several services with the same function
(24)     compare(maxProfitofBuyer, p_b (service_set));
(25)     //return the buyer pursues a better quality service
(26) end if
(27) for $j$ = 1 to $l$ do
(28)     $F$_s[$j$] = check the recommendation frequency of $s_j$;
(29)     if $F$_s[$j$] < LOW
(30)         change the service provider to update the
(31) service configuration;
(32) //then after the update, the service will be renewed;
(33)     end if
(34) end for
```

Algorithm 1: Trading between buyers and sellers.

Table 3: Quality of service (QoS) parameters for cloud services.

rateID	consumerID	serviceID	rateValue
int	int	int	float (calculated from the user's evaluation of the QoS attributes)

5. Experiments and Analysis

We produced a simulation program in Java to validate our reputation revision method. We simulated several service providers, which had different services with the same functions, as well as transaction behaviors of consumers. Our experiment comprised 500 consumers and four services, where each consumer rated the services they used after each transaction. The results were saved in the format shown in Table 3. To filter abnormal evaluations, our reputation revision method was used to cluster and mine the data in the transaction records after each round of transactions.

In our experiments, we tested two main hypotheses: (1) the consumer ratings of services follow a normal distribution; (2) the use of the approach described in Section 4 improves the accuracy of reputation ratings.

5.1. Distribution of Consumer Ratings.

To verify the service score distribution, the consumers were divided into three types (i.e., bad, right, and good) who gave different ratings to services in our experiment. The bad consumers gave a score

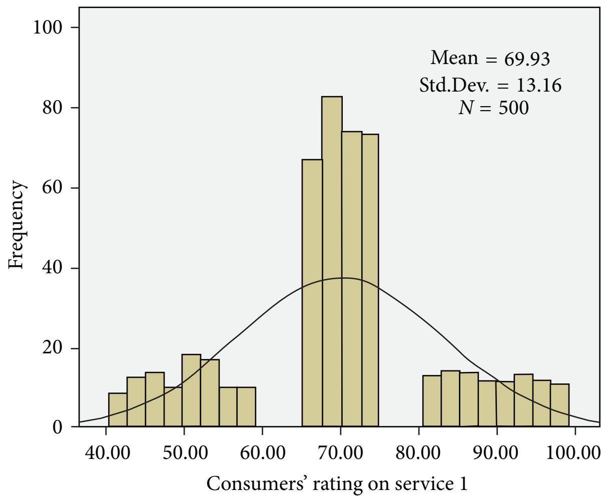

Figure 5: Distribution of rateValue for service 1 in the experiments ($N = 500$ users).

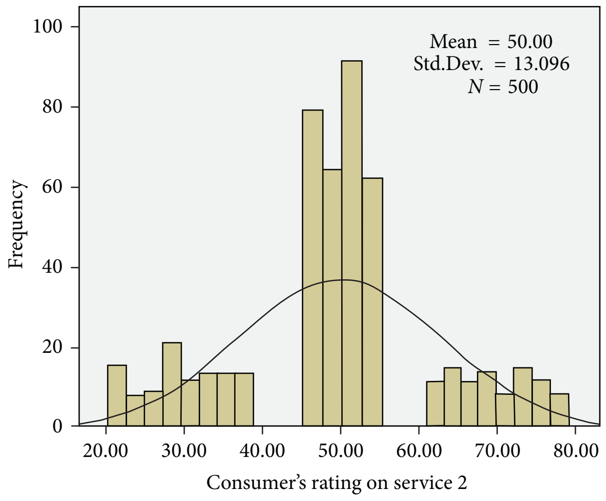

Figure 6: Distribution of rateValue for service 2 in the experiments ($N = 500$ users).

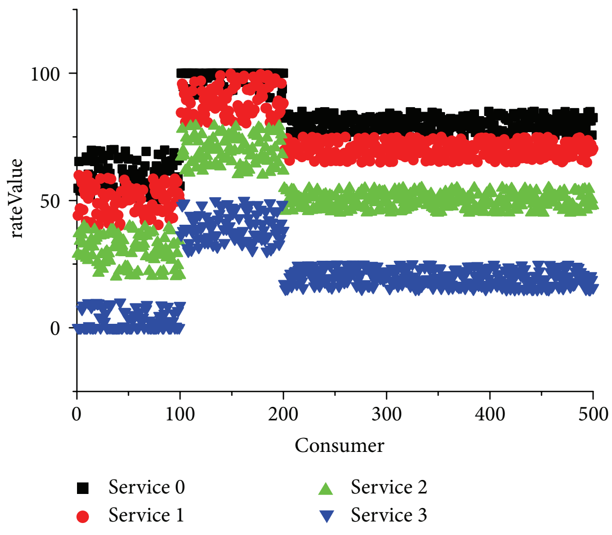

Figure 7: Ratings for four services given by 500 consumers.

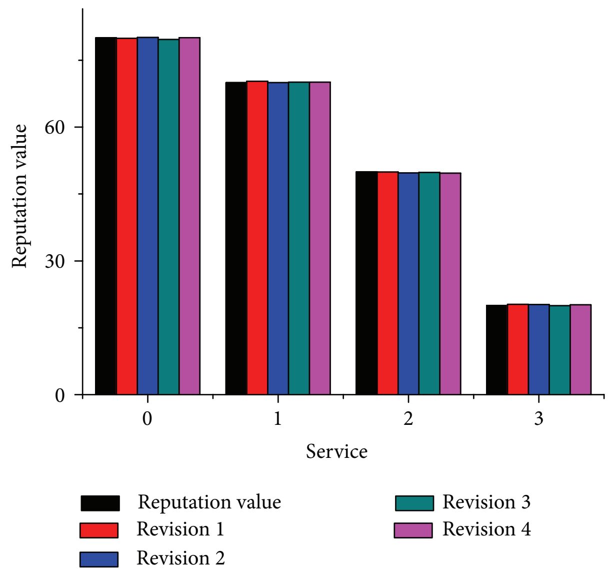

Figure 8: Four rounds of revised reputation.

below the service's QoS level and the good consumers gave a score higher than the QoS level. The right consumers gave a score around the QoS level. All of the consumers gave random scores from the corresponding interval.

The following QoS metrics were considered in the experiments: cost, response time, and execution time. The cost value was selected randomly from the range $[1, 10]\$$ and the response time was selected randomly from the range $[1, 1000]$ s. The execution time was set as the functional unit/single unit. The experimental data were analyzed using SPSS. Figures 5 and 6 show the score distributions for the two services.

The results of the data analysis showed that the distribution of the service rating scores was not a strictly normal distribution. However, we used the confidence interval (the formula proposed in Section 4) to identify the right consumer ratings and to filter the bad and good ratings. Some value intervals were empty because the consumer ratings were based on the QoS level of the service and the consumers' preferences.

5.2. Reputation Revision. In this experiment, we validated whether our method improved the accuracy of the service reputation ratings. In this experiment, four services provided the same function, but the QoS value belonged to different classes. The actual reputation of each service equaled the QoS value. We considered the revised reputation results for the four services after four cycles of revisions had been calculated. Figure 7 shows the ratings made by 500 consumers for each service they used. Figure 8 shows the results after four cycles of reputation revision.

Figure 7 shows that the four services had different rateValue levels and the ratings also reflected the three different types of consumers, who had distinct scoring trends. Figure 8 demonstrates the accuracy and stability of the reputation revision method. The experiment showed that the revision method could identify abnormal reputation ratings, which were filtered from the overall evaluations to improve the accuracy of the service.

6. Conclusions

In this study, we developed a reputation revision method for cloud services based on the confidence interval of a normal distribution of ratings and a market mechanism. To address the problem of abnormal evaluations, we used prior knowledge to distinguish between different types of consumers before filtering dishonest ratings and setting the preferences for consumers.

There are still some limitations in the dynamic provision of services and reputation management [26] for each distributed submarket. The dynamic provision of services will add more complexity to the processing of historical data. It will also be necessary to consider reputation management in the distributed submarkets.

Future research should investigate the development of a mechanism for the dynamic evolution of reputation ratings and the application of this method to large-scale service-oriented systems. The growing number of services and users means that the configuration parameters of services and the user base are changing constantly, so reputation evaluations should be capable of evolving. We also expect that this method could be deployed in a real cloud service application system. Further verification of this reputation revision mechanism will help to identify new problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61003035, U1204614, and 61370221 and in part by the Plan for Scientific Innovation Talent of Henan Province under Grant no. 124100510006.

References

[1] Z. Yan, Y. Chen, and Y. Shen, "A practical reputation system for pervasive social chatting," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 556–572, 2013.

[2] A. Satsiou and L. Tassiulas, "Reputation-based resource allocation in P2P systems of rational users," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 466–479, 2010.

[3] T. Chen, A. Bansal, and S. Zhong, "A reputation system for wireless mesh networks using network coding," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 535–541, 2011.

[4] B. Qureshi, G. Min, and D. Kouvatsos, "Countering the collusion attack with a multidimensional decentralized trust and reputation model in disconnected MANETs," *Multimedia Tools and Applications*, vol. 66, no. 2, pp. 303–323, 2013.

[5] Z. Li, H. Shen, and K. Sapra, "Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks," in *Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium (IPDPS '11)*, pp. 532–543, May 2011.

[6] L. Jiang, L. Ding, J. Liu, and J. Chen, "Reputation rating modeling for open environment lack of communication by using online social cognition," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 881–891, 2012.

[7] F. Faniyi and R. Bahsoon, "Self-managing SLA compliance in cloud architectures: a market-based approach," in *Proceedings of the 3rd international ACM SIGSOFT symposium on Architecting Critical Systems*, pp. 61–70, ACM, 2012.

[8] X. Wu, "A fuzzy reputation-based trust management scheme for cloud computing," *International Journal of Digital Content Technology and Its Applications*, vol. 6, no. 17, pp. 437–445, 2012.

[9] N. Yang, X. Chen, and H. Yu, "A reputation evaluation technique for web services," *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 329–334, 2012.

[10] A. Ghaffarinejad and M. K. Akbari, "An incentive compatible and distributed reputation mechanism based on context similarity for service oriented systems," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 863–875, 2013.

[11] H. Zhao, X. Yang, and X. Li, "An incentive mechanism to reinforce truthful reports in reputation systems," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 951–961, 2012.

[12] X. Su, M. Zhang, Y. Mu et al., "A robust trust model for service oriented systems," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 596–608, 2013.

[13] T. Kaszuba, A. Hupa, and A. Wierzbicki, "Advanced feedback management for internet auction reputation systems," *IEEE Internet Computing*, vol. 14, no. 5, pp. 31–37, 2010.

[14] X. Li, F. Zhou, and X. Yang, "Scalable feedback aggregating (SFA) overlay for large-scale P2P trust management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1944–1957, 2012.

[15] Z. Yan, P. Zhang, and R. H. Deng, "TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 485–506, 2012.

[16] M. Hawa, L. As-Sayid-Ahmad, and L. D. Khalaf, "On enhancing reputation management using Peer-to-Peer interaction history," *Peer-To-Peer Networking and Applications*, vol. 6, no. 1, pp. 101–113, 2013.

[17] K. Lu, H. Jiang, M. Li et al., "Resources collaborative scheduling model based on trust mechanism in cloud," in *Proceedings of the IEEE 11th International Conference on Security and Privacy in Computing and Communications (TrustCom '12)*, pp. 863–868, IEEE, 2012.

[18] Y. A. Kim and R. Phalak, "A trust prediction framework in rating-based experience sharing social networks without a Web of Trust," *Information Sciences*, vol. 191, pp. 128–145, 2012.

[19] C. Gui, Q. Jian, H. Wang, and Q. Wu, "Repeated game theory based penalty-incentive mechanism in internet-based virtual computing environment," *Journal of Software*, vol. 21, no. 12, pp. 3042–3055, 2010.

[20] O. Kussul, N. Kussul, and S. Skakun, "Assessing security threat scenarios for utility-based reputation model in grids," *Computers & Security*, vol. 34, pp. 1–15, 2013.

[21] P. Dong-Sheng, L. Chuang, and L. Wei-Dong, "A distributed trust mechanism directly evaluating reputation of nodes," *Journal of Software*, vol. 19, no. 4, pp. 946–955, 2008.

[22] J. Gorner, J. Zhang, and R. Cohen, "Improving trust modelling through the limit of advisor network size and use of referrals," *Electronic Commerce Research and Applications*, vol. 12, no. 2, pp. 112–123, 2013.

[23] S. Zhao, G. Wu, G. Chen, and H. Chen, "Reputation-aware service selection based on QOS similarity," *Journal of Networks*, vol. 6, no. 7, pp. 950–957, 2011.

[24] Y. Cui, M. Li, Y. Xiang, Y. Ren, and S. Cesare, "A QoS-based fine-grained reputation system in the grid environment," *Concurrency Computation Practice and Experience*, vol. 24, no. 17, pp. 1990–2006, 2012.

[25] M. Hussin, Y. C. Lee, and A. Y. Zomaya, "Reputation-based resource allocation in market-oriented distributed systems," in *Algorithms and Architectures for Parallel Processing*, pp. 443–452, Springer, Berlin, Germany, 2011.

[26] Z. Bankovic, D. Fraga, J. Manuel Moya et al., "Improving security in WMNs with reputation systems and self-organizing maps," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 455–463, 2011.