Reputation-aware Service Selection based on QoS Similarity

Shenghui Zhao\textsuperscript{1,2}, Guoxin Wu\textsuperscript{2}
\textsuperscript{1}Department of Computer Science and Technology, Chuzhou University, Chuzhou, China
\textsuperscript{2}School of Computer Science & Engineering, Southeast University, Nanjing, China
Email:zsh@chzu.edu.cn, gwu@seu.edu.cn

Guilin Chen, Haibao Chen
Department of Computer Science and Technology, Chuzhou University, Chuzhou, China
Email:glchen@chzu.edu.cn, chb@chzu.edu.cn

Abstract—For the up-and-coming computing models like as cloud computing, service is the standard package for meeting all kinds of consumers’ requirements. Web Services are the concrete implement of the service. When users request and consume Web Services, services’ reputations will play a vital role in users’ selection. A gradually adjusting reputation evaluation method of Web Services is proposed based on eliminating the collusive behaviors of consumers step by step, and a reputation-aware model for service selection is designed. In order to adjust reputations, QoS similarity is computed firstly according to the differences between advertised QoS from service providers and delivered QoS from service consumers’ evaluation, next, current reputation is attained; then the consumers are sorted based on reputation using clustering algorithm and the potential collusive consumers are mined using association rules algorithm; finally, the updated reputation is recalculated and saved in the reputation center included in the model. The experimental results show that the model can identify the malicious consumers and improve the exact rate of reputation evaluation and success rate of service selection.

Index Terms—Web Service, quality of service (QoS), reputation update, clustering algorithm, collusive consumers

I. INTRODUCTION

With the widespread of SOA, Web Services has become the main computing paradigm across Internet, new computing patterns are springing up such as cloud computing and CPS (Cyber Physical Systems) etc. A Web Service is a self-described and self-contained application that uses standard Internet technologies to interact with other Web Services, which can be published and accessed through the web. At present, many corporations and organizations have implemented their core application through buying the Web Services on Internet. For example, salesforce.com provides ERP service for users. Along with the maturation of service market, more and more service providers can provide the same or similar service, how to rationally select satisfied service has been turned into one of the key problems in Web Services research fields.

When service requestors select required services among many services with similar functionality, services’ non-functional properties is an important considerable criterion, such as QoS (Quality of Service), reputation, etc. Generally speaking, QoS of Web Service is described by response time, reliability, availability, security and execution cost and so on. In the early service transactions, QoS information was published by service providers, but it was not always exact and up-to-date. For the interest of ensuring the veracity of QoS properties, it should be a direct and valid method to appraise the QoS by requestors after invoking the Web Service. These values can be acted as the references for subsequent consumers to select the service. Many researches on service selection adopt this scheme.

However, in the practical transactions, some feedbacks about QoS are falsity information due to the vicious estimation aiming at service providers. Thus, relying only on feedback estimation of QoS can not provide accurate methods for service selection. Reputation based service selection methods were proposed later, most of which were reputation evaluation on the basis of appreciable QoS after invoking a service, and then computed predicted reputation integrating multi historical values and current value. Above methods can wipe off influence of little vicious users at a certain extent and improve the success rate of service selection. But, the community collusions may be occurred among consumers or among consumers and providers, services' reputation may be either lower or rose up which leads to distortion of reputation.

This paper discusses service selection based on reputation, in which distinguishes and filters out the collusive consumers through collusive behavior analysis methods. Then these collusive consumers' ratings are ignored, and decrease the influence of the malicious consumers, which can improve the veracity of reputation and the success rate of service selection. The rest of the paper is organized as follows. In section 2, we introduce related work. Section 3 proposes a method for Web Service reputation evaluation. A model for service
selection is set up in Section 4. Next is experimental analysis. At last, we conclude the paper.

II. RELATED WORK

It is essential to acquire the QoS information when service selection is depending on QoS. [1] presented that both service selection and composition were QoS-aware, the QoS was measured by monitoring system according to service operations. An approach for measuring quality of Web Services based on the superposition of uncertain factors was proposed, and a judging method for determining priorities among Web Services, which can help users select satisfied service [2]. A dynamic and QoS-driven model for service selection was proposed in [3], and the dynamic QoS data were computed according to users’ feedback. In [4], the QoS attributes’ data obtained from service providers was revised, and feedback similarity came from service consumers was used to weight QoS data’ trustworthiness, that strengthened the accuracy of the service selection. In [5], service-level agreements were discussed in order to set the penalties over the lack of QoS for web services. It ensured that the trustworthiness of a service-oriented environment relies on reliable QoS monitoring in certain sense.

Although QoS based service selection is essential, due to the services’ marketability, it is hard to avoid the dishonest service providers. So, service selection must be trust or reputation based [6], that can assure service’s trustworthiness. In [7], Yao Wang et al. reviewed and concluded the service selection’ criteria, and presented that it was necessary to implement service selection depending on trust and reputation. The authors in [8] suggested a framework of service selection based on reputation in a semantic network. The reputation was computed by different service consumers. In [9], Malik et al. had proposed a model to compute the reputation of a web service in accordance with the personal evaluation of the previous users. The characteristic of this method was the credibility of the users of evaluating services has been taken into account. If the rater tried to provide a fake rating, then its credibility would be decreased and the rating of this user would become less important in the reputation of the web service.

Obviously, QoS can help consumers select the service with high quality, and reputation has been used to make consumers select the service providers which honestly offer the service with advertised QoS. Making use of reputation, consumers can find or select secure, reliable and trusted Web Services. So, the service quality’s reputation is vital important to select the genuine service required by the consumers. In [10], Maximilien and Singh designed a multi-agent framework based on ontology for QoS. The users’ ratings which depended on the different qualities satisfied varied consumers’ trust requirement used for computing the reputation of the web service, and it would be the selection criterion. That was dynamic selection.

Ping Wang et al.[11] expressed an idea that aggregating previous assessment records (bodies of evidence) via consumers’ feedbacks and witness of network referrals to derive a more objective reputation score on the specific service. Then two factors was defined, confidence degree and support degree based on evidence theory, to enhance the discrimination of the quality of existing evidence to help providers avoid malicious assessment. In [12], service providers’ reputations were figured out through applying the current reputation and historical data with various weights. According to providers’ reputation and services’ reputation, a method for measuring service providers’ trust was proposed. By ranking the trust value, consumers can select more trusted service. In [13], the authors developed a framework aiming to select Web Services based on the trust policy expressed by the users. The framework allowed the users to select a web service matching their needs and expectations.

In above-mentioned literatures about service selection, some only proposed the models of applying reputation, and some gave the rating methods on reputation, but they lacked of controlling the situations of providing falsity reputation information. That is, there are only few researches on consumers’ collusion and deception. Although some researches [14][15] have considered researches on consumers’ collusion and deception. Although some researches [14][15] have considered collusion among consumers, their analysis objects and methods are different from our paper. Moreover, most literatures didn’t give solutions on integrality of reputation data.

For the sake of solving above problems, in our reputation-aware service selection model, the reputation is computed based on the similarity of service quality’s attributes, and a method for collusion behavior analysis is proposed. Besides, while constructing a system reputation model, we take into account reputation storage and security, which can provide a relative secure and trusted service selection scheme for consumers.

III. A METHOD FOR RATING WEB SERVICE

The evaluation method is based on following suppositions:
1) One service provider offers one service only, and the provider’s reputation can be apperceived by its’ service reputation.
2) The reputation published to UDDI by service providers is authentic.
3) The reputation center is trustable. It can be acted as a broker for service consumers and providers’ behaviors.
4) If one service consumer selects a service, it must trust in the service provider.
5) If the consumer is honesty, its evaluation is honesty too.

A. Computation of Web Service’s Reputation.

Supposing that a Web Service $S_j$ has $m$ attributes of QoS, what is expressed as $(q_{1j}, q_{2j}, ..., q_{mj})$. For any user $u_i$, its invocation to $S_j$’s attributes is represented as $(q'_{1i}, q'_{2i}, ..., q'_{mi})$. The advertised QoS of $S_j$ is shown as $(A_{d}q_{1j}, A_{d}q_{2j}, ..., A_{d}q_{mj})$. After $S_j$ being invoked by
\( u_j \), its feedback rate on \( s_j \)'s QoS is denoted as \(( eval_{-} q_i^j, eval_{-} q_i^n, ... , eval_{-} q_m^n)\).

**Definition 1** Similarity of service \( s_j \)'s quality \( Sim^j \)

After current user \( u_i \) invoking service \( s_j \), \( u_i \) will give a new value on QoS' attributes. The similarity degree \( Sim^j \) can be figured out in (1).

\[
Sim^j = 1 - \sqrt{\frac{\sum_{i=1}^{m} (eval_{-}q_i^n - Ad_{-}q_i^n)^2}{m}}
\]  

(1)

\( m \) is the number of QoS attributes.

**Definition 2** Service \( s_j \)'s current reputation \( Rep^j_{cur} \)

\( s_j \)'s current reputation is the newest reputation after this invoking. \( Rep^j_{cur} \) is computed using (2) which is figured as Fig. 1.

\[
Rep^j_{cur} = 1 - \sinh(1 - Sim^j).
\]  

(2)

Thereinto, \( \sinh(x) = (\exp(x) - \exp(-x))/2 \).

![Figure 1. Result of Function \( \sinh(x) \)](image)

It can be seen from Fig. 1, when advertised QoS is closer to received QoS, the higher reputation is gained. Especially, if \( Sim^j \) is equal to 1, \( Rep^j_{cur} \) can achieve highest value 1. When \( Sim^j \) is less than 0.2, \( Rep^j_{cur} \) is nearly 0. This is very similar to the actual application occasions. So we adopt the function in formula (2).

But current reputation had something to do with rater's credibility degree. If rater is trustful user, its reputation is trustworthy and can be contained in the global reputation. Otherwise, its reputation is not trustful and will be ignored in the global reputation.

**B. Reputation Update and Adjustment**

When \( s_j \) is invoked at each time, \( s_j \)'s current reputation \( Rep^j_{cur} \) can be computed using formula (2). However, for other users, they not only use the last result, but also consult historical data. A reputation center being involved in the service selection model is designed in Section 4. Reputation center plays the role of computing and keeping each service's global reputation. Hence, after each time invoking \( s_j \), it should update the stored \( s_j \)'s global reputation.

Assuming parameter \( \delta \) is the difference between received QoS and advertised QoS, that is,

\[
\delta = \sum_{i=1}^{m} (eval_{-}q_i^n - Ad_{-}q_i^n)/m \quad \text{If } \delta \quad \text{is larger than 0, it can be explained that received QoS is prior to advertised QoS. Then the reputation will be increased apparently; otherwise it will be decreased.}
\]

**Definition 3** Service \( s_j \)'s global reputation \( Rep^j_{glo} \)

\( s_j \)'s global reputation \( Rep^j_{glo} \) is associated with all historical reputations and current reputation. After each invoking, \( Rep^j_{cur} \) and historical data are updated and \( Rep^j_{glo} \) is adjusted using formula (3).

\[
Rep^j_{glo} = Rep^j_{cur} \times (0.9 + 0.1 \times \exp(\delta))
\]  

(3)

In (3), \( Rep^j_{cur} = \sum_{i=1}^{\gamma_i} rep^j_{cur} \times \gamma_i / \sum_{i=1}^{\gamma_i} \gamma_i, \quad \gamma_i = \lambda^d_{-} \)

\( rep^j_{cur} \) is the \( i \)th current reputation of service \( s_j \) computed by formula (2). \( \gamma_i \) is the aging factor for the \( i \)th service reputation, \( \lambda \in [0, 1] \). A smaller \( \lambda \) means only recent reputations are included and a larger \( \lambda \) means more reputations are included. \( d_i \) is the time interval of between last rating time and current time. For instance, when using current reputation, \( d_i \) almost equals to 0, so \( \gamma_i \) is 1. That is, the current reputation has not aged.

**C. Reputation Storage**

In general, there are three places to store reputation information, rater (service consumer), ratee(service providers) and a third party(reputation center). This paper stores the information in ratee and a third party, respectively. The advantage of saving the global reputation in ratees is that consumers can find the satisfied service's reputation from providers at the time of reputation center collapsing. Due to computing global reputation in reputation center, each service's rating information should be saved. A database is created in reputation center to save rating information. The storage format includes five items, listed in Table I.

<table>
<thead>
<tr>
<th>Table I. Storage Format of Reputation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rater ID (UID)</td>
</tr>
<tr>
<td>Rate0001</td>
</tr>
</tbody>
</table>

When a consumer finds the satisfied service providers, he will query the providers' reputation in either reputation
center or providers itself. In order to prevent the providers to tamper with the saved reputation, the reputations will be handled in particular. The method is adopting the ReertX[16] idea to save the reputation information, where each service's reputation is put in one certificate, so all reputations are linked as a body to avoid the provider juggling with the reputation. After each global reputation is updated, it will be signed digital and stored in reputation center and sent to ratee. When users take out the reputations from the providers, they will use public key published by reputation center to decrypt the messages. Of course, public key has been already delivered to all consumers by broadcast, and it can be transmitted among consumers too.

D. Algorithm for collusion behavior analysis

After service transactions, honest users will give fairness and objectivity valuation. But the malicious consumers will give unpractical estimation, like as rising up or playing down the reputation intentionally, as well as collusive users. If the proportion of collusive users is higher, it will affect the service providers' reputation more greatly. Therefore, reputation model should be capable of identifying the collusive users and reduce the negative influence.

For the sake of finding the collusive consumers, we design an algorithm named as CBA (Collusive Behavior Analysis) which has two main operations: classification and mining. We will use k-means cluster algorithm [17] to create multi clusters and find the correlations among consumers adopting the association rule mining algorithm.

The CBA algorithm steps are as follows:

Step 1 Take out all records of service s_j from the database in reputation center, and sort the reputation values into three clusters (marked as 0,1,2) using k-means cluster algorithm. The 0th cluster denotes the class with the lowest reputation, and the 2th cluster represents the class with the highest reputation. Average the reputation distributed in the ith cluster, and makes the mean as the references to analyze the collusion.

Step 2 Owing to the reputations in the class of 0th clusters are all lower generally, which express that all consumers adopting the association rule mining algorithm.

TABLE II. TRANSACTION RECORD

<table>
<thead>
<tr>
<th>TID</th>
<th>UID</th>
<th>SID</th>
<th>RepValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u1</td>
<td>s1</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>u2</td>
<td>s2</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>u2</td>
<td>s1</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>u2</td>
<td>s1</td>
<td>0.65</td>
</tr>
<tr>
<td>5</td>
<td>u3</td>
<td>s2</td>
<td>0.55</td>
</tr>
<tr>
<td>6</td>
<td>u1</td>
<td>s2</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>u2</td>
<td>s2</td>
<td>0.85</td>
</tr>
<tr>
<td>8</td>
<td>u1</td>
<td>s1</td>
<td>0.86</td>
</tr>
<tr>
<td>9</td>
<td>u3</td>
<td>s1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Step 3 Applying the association rules algorithm, on given support degree, the largest frequent items can be mined, in which there are different users.

Step 4 The different users in the largest frequent items may be looked as collusive community. So, if the user exists in the community, it will be marked as dishonest user and set h_i=0; otherwise, set h_i=1.

After detecting the collusive users, recalculate rep^i\_s_j, viz. rep^i\_s_j = h_i \times rep^i\_s_j. So, when h_i=0, the current reputation is zero, too.

For example, there are two services, s1 and s2. Their 30 transaction records are listed as Table II.

Based on the above algorithm and using k-means cluster algorithm, we extract all s1 records and s2 records and divide them into three clusters, respectively. The results are shown in Table III and Table IV.

TABLE III. DIVIDE S1 RECORDS INTO THREE CLUSTERS

<table>
<thead>
<tr>
<th>TID</th>
<th>UID</th>
<th>SID</th>
<th>RepValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u1</td>
<td>s1</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>u2</td>
<td>s2</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>u2</td>
<td>s1</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>u2</td>
<td>s1</td>
<td>0.65</td>
</tr>
<tr>
<td>5</td>
<td>u3</td>
<td>s2</td>
<td>0.55</td>
</tr>
<tr>
<td>6</td>
<td>u1</td>
<td>s2</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>u2</td>
<td>s2</td>
<td>0.85</td>
</tr>
<tr>
<td>8</td>
<td>u1</td>
<td>s1</td>
<td>0.86</td>
</tr>
<tr>
<td>9</td>
<td>u3</td>
<td>s1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

TABLE IV. DIVIDE S2 RECORDS INTO THREE CLUSTERS

<table>
<thead>
<tr>
<th>TID</th>
<th>UID</th>
<th>SID</th>
<th>RepValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u1</td>
<td>s1</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>u2</td>
<td>s2</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>u2</td>
<td>s1</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>u2</td>
<td>s1</td>
<td>0.65</td>
</tr>
<tr>
<td>5</td>
<td>u3</td>
<td>s2</td>
<td>0.55</td>
</tr>
<tr>
<td>6</td>
<td>u1</td>
<td>s2</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>u2</td>
<td>s2</td>
<td>0.85</td>
</tr>
<tr>
<td>8</td>
<td>u1</td>
<td>s1</td>
<td>0.86</td>
</tr>
<tr>
<td>9</td>
<td>u3</td>
<td>s1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

© 2011 ACADEMY PUBLISHER
From Table III and Table IV., we can see, users' list (u3,u4,u2) and (u2,u3,u5) derived from cluster0 are made up of two lines. If there are more services, they will formed as 

\[
\{u3,u4,u2, u2,u3,u5\}
\]

in which collusive consumers could be mined using association rule mining algorithm.

IV. SERVICE SELECTION MODEL OF REPUTATION-AWARE E

Service selection model based on reputation is designed and shown as Fig. 2. It includes two roles: service providers, service requestors, and a data center: UDDI, as well as three agents: discovery agent, selection agent and rating agent. Discovery agent helps service consumers to find services meeting requirements, and selection agent selects the service with highest reputation for consumers from the satisfied services. The purpose of rating agent is to evaluate the newest reputation of service providers and update the global reputation.

![Figure 2. Model of Service Selection](image)

The process of service selection is as follows.

1) If a service provider joins the system, it will publish its advertised QoS to UDDI acquisiently.

2) Service requestors (viz. consumers) send their requirements which contain functional and nonfunctional descriptions to discovery agent.

3) Discovery agent queries the UDDI. If UDDI has the satisfying services, it returns the results to discovery agent.

4) Discovery agent submits the results to selection agent.

5) Selection agent will query the reputations of all matched services to reputation center.

6) Selection agent sorts the reputations of all satisfied services, and returns service with the highest reputation to requestor.

7) Service provider and requestor carry out their transaction.

8) After using service, consumer evaluates the aware quality of service, and sends information back to reputation center.

9) Rating agent sends request to UDDI for promised QoS of service provider. Then updates the service' global reputation according to formula (2) and (3).

10) Reputation center makes digital signature for the reputation and feed it back to the provider.

CBA algorithm will be executed by reputation center after a period of time. The results can be used in 9).

V. EXPERIMENTATION

In order to validate our reputation model, we develop a simulation program written in Java. It simulates multi service provider and consumers' transaction behavior. The transaction results are saved like as the format listed in Table I. For the purpose of finding the collusive users, CBA algorithm is employed to make clustering and mine data for the transaction records at regular intervals. As time goes on, collusive consumers will gradually be steady. So the diminished reputations also tend towards stability. In experiments, the records being used are in a fixed time of transactions.

A. Experimental Environment

In our simulation environment, the number of service providers is 100, and the number of service consumers is 60. For the convenience of experiment, the kind of services offered by providers is 1. Each service's QoS includes availability, response time, reliability, security and cost. Each service's attributes has four levels: bad, ordinary, good, and excellent. We assumed that there are 10% excellent, 20% bad, 30% good and 40% ordinary among the 100 providers.

However the service level is, the collusive consumers will give bad evaluation, and honest consumers will give authentic estimation. The appraisal is like as Table V.

![Table V. Estimation Range from Consumers](image)
At the beginning of experiment, assumed all users are honest and all services' initial reputation are 0.6. The value of execution transaction is 0.6, meaning that if reputation is greater than or equal to 0.6, a transaction will be executed. At the same time, the successful transaction refers to the reputation greater than a threshold. Given threshold 0.7, viz. if the estimation value is no less than 0.7, the transaction is successful. We will take success rate of service as a validity measurement of reputation model. Success rate of service is defined as the ratio of the number of times to the number of total transactions. When finding maximum frequent items, we set the support degree 0.9.

B. Experimental Data

In order to examine the validity and veracity of reputation computing in service selection model, we use rate of service success and exact rate of reputation to measure, respectively. Comparison is implemented in the two situations of considering collusion and no considering it.

(1) Comparison of Success Rate

At first, we study the success rate when the ratio of collusion (RoC) varies from 10% to 40% increased by 10%. The results of considering collusion and no considering collusion are shown in Fig.3 and Fig.4.

<table>
<thead>
<tr>
<th>RoC</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>0.8</td>
</tr>
<tr>
<td>20%</td>
<td>0.7</td>
</tr>
<tr>
<td>30%</td>
<td>0.6</td>
</tr>
<tr>
<td>40%</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Figure 3. Success Rate of Considering Collusion under different RoC

Figure 4. Success Rate of No Considering Collusion under different RoC

From Fig.3 and Fig.4, the success rate reduces evidently as the RoC increasing, it can be explained that along with the number of round grew higher, success rate goes to steady. When the RoC is 10%, both success rates are approaching to 90%, what can be explained the effect of collusion is not distinctness. But, when the RoC arrives at 40%, the success rate of considering collusion is higher than no considering collusion. Obviously, considering collusion has better effect for improving success rate. If the ratio of collusive users is bigger, it will have a larger effect on the service providers' reputation.

<table>
<thead>
<tr>
<th>service level</th>
<th>Proportion of service providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>10% 20% 30% 40% 50%</td>
</tr>
<tr>
<td>Good</td>
<td>20% 20% 20% 20% 20%</td>
</tr>
<tr>
<td>Ordinary</td>
<td>50% 40% 30% 20% 10%</td>
</tr>
<tr>
<td>Bad</td>
<td>20% 20% 20% 20% 20%</td>
</tr>
</tbody>
</table>

Secondly, we compare the success rate of different ratio of excellent services under the circumstances of 20% RoC. The ratio of excellent services (RoE) is listed in Table VI. When the ratio of excellent service changes from 10% to 50%, the rates of success are displayed in Fig.5.

(2) Exact Rate of Reputation

A service's exact rate is represented as:

\[ \text{exactRate} = 1 - \frac{\text{rep}_{\text{true}}}{\text{rep}_{\text{ph}}} \]

The experimental results are shown in Fig.6. Each value is the average of exact rates in each round. With the increase of rounds, about 500 rounds later, the exact rate goes to stable. The exact rates of considering collusion and no considering collusion are about 92% and 72%, respectively. Above result illustrates our reputation model considering collusive behavior can eliminate the influence of collusion effectively.
In order to differentiate reputation-aware selection from QoS-aware selection, we set the experiment of the QoS-aware service selection only relying on the evaluated QoS. Table VII is hold good. The threshold of QoS is set to 0.6, that is, if the evaluated QoS is 0.6, the transaction succeeds. The colluding consumers are filtered out using our CBA algorithm and total QoS value but not global reputation is computed. Total QoS value is the average of received QoS. The service selection is based on the ranking QoS, obviously, the service with highest QoS will be selected.

The result is shown as Fig.7. Compared with Fig.3, the rate of success is lower than 15% in general. That illustrates service selection of reputation-aware is superior to QoS-aware.

VI. CONCLUSION

This paper introduces a reputation model considering collusive consumers. We get the current reputation by utilizing the similarity between advertised QoS from service providers and delivered QoS from consumer’s evaluation, then update the global reputation and save them into reputation center and service providers. At the same time, in order to prevent providers tampering with reputation, we use the digital signature. Experimental results show that the success rate of transaction considering collusion is higher than no considering collusion.

In order to find collusive consumers, we make use of k-means cluster algorithm to classify the consumers, and use association rule algorithm to mine collusive consumers, then adjust the service reputation through eliminating the collusive consumers gradually. In fact, collusion exists not only in consumers, but also exists between consumers and providers. Because of profits, they have enthusiasm to make collusion. How to reduce the second collusion is another challenge.

ACKNOWLEDGMENT

This research was partially supported by Anhui Educational Department Natural Sciences Project (KJ2010A251).

REFERENCES


Shenghui Zhao Ph.D candidate of Southeast University, associate Professor of Chuzhou University. Her major research fields include network security, Web Services and distributed computing.

Guoxin Wu Professor of Southeast University, doctoral supervisor. His major research fields include trust network, distributed computing and network security.

Guilin Chen Professor of Chuzhou University. His research interests include distributed computing, pervasive computing and virtualization.

Haibao Chen Teacher of Chuzhou University. His major research fields include trust and Cloud computing.