Applied Mathematics & Information Sciences
An International Journal

A Trust Evaluation Mechanism for Collaboration of Data-Intensive Services in Cloud

Longtao Huang[1], *Shuiguang Deng*[1,*], *Ying Li*[1], *Jian Wu*[1], *Jianwei Yin*[1], *and Gexin Li*[2]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[2]Institute of Information System Integration, Wenzhou University, Zhejiang Wenzhou 325025, China

Received: 27 Oct. 2012, Revised: 20 Dec. 2012, Accepted: 1 Jan. 2013
Published online: 1 Feb. 2013

Abstract: Trust and reputation for services emerges as an important issue in cloud computing. Since data-intensive services in cloud have been used in more and more fields, trust evaluation for collaboration of services meets more challenges. There are not only logical dependencies but also data dependencies among partner services when data-intensive services take part in collaboration. This paper proposes a novel trust evaluation method for collaborations of data-intensive services. It considers not only the trust for individual partner services and the explicit trust relation among partner services that have logical dependencies for each other, but also the implicit trust relation implied in data-dependencies among services. A serial of experiments, using the simulation tool NetLogo, are carried out to compare the evaluation results between the proposed method and the method without data-dependency consideration. The result shows that taking consideration of the data-dependency trust improves the accuracy of trust evaluation to a great extent.

Keywords: Trust, Data-Intensive, SOA, Cloud Computing

1. Introduction

The emergence of cloud environments has made a big movement towards the intensive, large scale specialization. At the same time, it brings about not only convenience and efficiency problems, but also great challenges on data security and privacy protection [1]. To decide whether a cloud service is trustworthy has become an important issue. With the increasing number of services in cloud, collaboration of simple services to fulfillment complex requirements has become into realities. As the partner services of collaboration come from different providers, it is essential to decide whether the collaboration is trustworthy before it is put into use. At present, there is a great deal of effort devoted to the issue of trust evaluation for service collaboration.

The explosion of data and information has been recognized in recent years. Generated data is growing too fast to store and handle it as before. Emerging cloud-based infrastructures for storage have been widely accepted as the next-generation solution to address the data proliferation and the reliance on data [2]. Then data-intensive services in cloud have been used in more and more fields. The emergence of data-intensive services in cloud also brings new challenges for trust evaluation for service collaboration. Traditional trust evaluation methods for service collaboration always evaluate the trust of each partner services and sum up the values according to the collaboration process. These methods may not suitable for service collaboration of data-intensive services. When date-intensive services cooperate with others, they may exchange more data directly and have more data-dependencies on each other. Thus, traditional trust evaluation methods for service collaboration are not suitable any more as the trust relation implied in data-dependencies between partner services has great influence on the trust of the whole collaboration of data-intensive services. In some cases, although all the partner services in service collaboration have high-level trust from the perspective of the composer, the collaboration may leads to a failed execution because some two services with a data-dependency do not trust each other.

Consider such an abstract scenario of collaboration of data-intensive services where a partner service r accepts data from t as input, i.e., r and t communicate with each other directly and r depends on the data from t. If r doesn't trust t completely according to its historic interaction records and thus it is not willing to accept the

* Corresponding author e-mail: dengsg@zju.edu.cn

data from t, then the collaboration will fail in execution even the coordinator has high-level trust both in r and t. The scenario indicates that trust in data-dependency between services is an important factor that should be considered for trust evaluation for collaboration of data-intensive services.

However, most of the existing work [3,4] focused on the trust of partner services and the explicit trust between services with logical dependencies, but neglected the data-dependency trust; thus they leaded to an inaccurate trust evaluation for collaboration of data-intensive services. This paper proposes a novel trust evaluation method for such kind of service collaboration with the consideration on the trust evaluation for data-dependencies between partner services. We also conduct a serial of experiments to verify the proposed method. As trust propagation does not manifest itself as a physical phenomenon in nature, but only exists on the mental and cognitive level, it is therefore difficult to assess whether computational models for trust propagation are adequate and reflect the way people reason about trust [5]. In order to check the accuracy of the proposed method in a real-world-like environment, we use NetLogo [6] to generate many agents with services to simulate the trust community and then compare the trust evaluation between the proposed method and the method without data-dependency consideration. The result indicates that taking consideration of the data-dependency trust improve the accuracy of trust evaluation to a great extent. The key contributions of this paper can be summarized as follows:

1) It proposes a formal trust model for collaboration of data-intensive services with the consideration on the data-dependency trust.

2) It presents a mechanism to evaluate the trust between partner services based on trust transitivity.

3) In order to verify the accuracy of the proposed trust model and evaluation method, it establishes a real-world-like trust community to conduct a serial of simulation experiments with NetLogo.

The rest of this paper is organized as follows: Section 2 gives some related work. Section 3 introduces the trust model for collaboration of data-intensive services. Section 4 presents the process of trust evaluation based on the proposed trust model. Section 5 carries out a serial of experiments. Finally, Section 6 concludes the work and gives the future direction.

2. Related Work

At present, there are many trust and reputation mechanisms proposed to decide whether a service is trustworthy [7,8]. Some trust models are based on statistical techniques. Jsang etc[7] introduced a trust model with help of beta probability density functions. And in [9], Nguyen etc proposed a trust model base on Bayesian network which is a modern statistic method to calculate the probability of a hypothesis under different conditions. Some other trust models are based on AI method. Malik etc[10] introduced a trust evaluation method using Hidden Markov Model, HMM is used to predict the reputation of a service provider when the feedback ratings are insufficient. Nepal etc[11] presented a trust management framework base on fuzzy set theory. They proposed a fuzzy trust data model and a fuzzy linguistic query model, so that consumers can express their queries without knowing the underlying data models. There are also some trust models taking consideration of other factors of services such as capability of services, security strategy. Li and Tian[12] proposed a capability enhanced trust model. The capability, security, and feedbacks are evaluated in an integrated manner to decide if a service is trustworthy. Zhao and Varadharajan[13] proposed a comprehensive trust management approach for web services that covered the analysis/modeling of trust relationships and the development of trust management layer in a consistent manner. In [14], Kovac and Trcek presented an abstract trust model that applies complementary qualitative methodology which addresses the core of trust as socio-cognitive phenomenon. The model complements existing quantitative methodologies and is applied in the web services environment that enables trust management in SOAs.

The trust models above are mainly for single services. For the trust evaluation for service compositions, there are also some works toward it. Nepal etc[15] developed a method of distributing reputation that was received by a composition to its partner services. This method enabled that partner services could get their reputation according to their contribution, and a partner service was neither penalized nor awarded for the bad and good performances of other partner services, respectively. So this method is a fair distribution. Hamdi Yahyaoui [16] presented a trust-based game theoretical model for services collaboration. The devised model is an application of the generic model about tasks allocation for agents. They provided a trust-based game in which the objective is to assign tasks to services in a way that maximizes the likelihood of performing successfully these tasks. Paradesi etc[17] proposed a framework Wisp to evaluate the trust for compositions. They adopted a formal model for trust in a service. Then they evaluated trust for compositions from trust models of partner services based on the types of composition flows. They mainly discussed 4 basic types of flows: sequence flow, concurrent flow, conditional flow, and loop. Compared to the work [17], we evaluate trust for composition not from the types of flow but from the logic paths with the consideration of data dependencies. Furthermore, Paradesi only considered the trust of partner services to evaluate the trust for compositions and didn't consider the trust relation between partner services that interact with each other with data-dependencies. And most of the existed methods didn't consider this important factor, which may

cause unreasonable evaluation for collaboration of data-intensive services. Our work considers not only the trust for partner services but also the trust elation between partner services that has data-dependencies.

3. Trust Model

In this section, we formally describe the trust model for service collaboration of data-intensive services, which considers the implicit trust relation of data-dependencies in addition with trust for partner services and logic-dependencies among partner services.

3.1. The Collaboration Model

We import the concept of Service Collaboration Graph with the abbreviation SCG to represent service collaboration. It can be defined formally as follows.

Definition 1 (SCG): SCG is a directed graph to illustrate the services relations in a collaboration and it can be modeled as a 4-tuple $SCG(S, N, L, D)$ where:

(1) S is the set of partner services in the collaboration.

(2) N is the set of logic nodes that control the execution logic of the collaboration and each element can be any one of XOR, OR and AND.

(3) L is the set of logic dependencies between services.

(4) D is the set of data dependencies between services.

Definition 2 (Logic Dependency): A Logic Dependency defines the order of execution among services.

At present, we mainly consider the following types of logic dependencies:

Sequence: A partner service is executed after the completion of another partner service. For example, W_7 and W_8 are executed in sequence as shown in Figure 1.

AND-split: A logic node that makes a single process into multiple processes which can be executed in parallel, thus allowing partner services to be executed simultaneously or in any order. For example, W_4 and W_6 can be executed in parallel after the AND node.

AND-join: A logic node where multiple parallel processes converge.

XOR-split: A logic node that makes a single process into multiple processes and only one can be selected to be executed. Note that OR-split is a similar logic but differs in the number of processes to be selected ranges from 0 to the total number of processes.

XOR-join: A node where two or more alternative branches come together without synchronization.

Loop: The partner services may be executed several times repeatedly. Here we suppose that the number of iterated times is fixed.

Definition 3 (Data Dependency): A Data Dependency defined between two services indicates that the output data/message from one service is used as an input of another. The relation of date dependency is a basic characteristic of collaboration of data-intensive services.

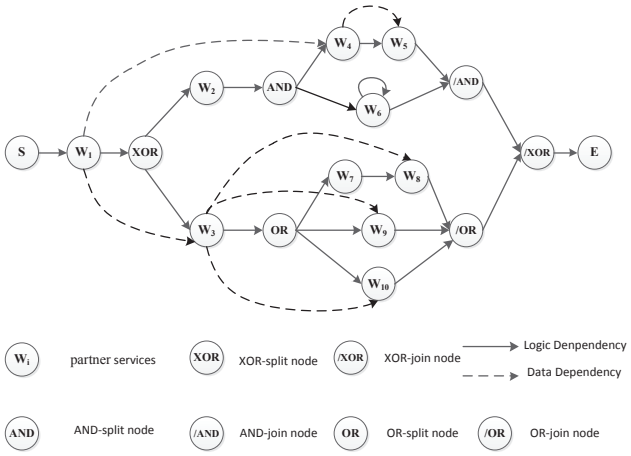


Figure 1: A SCG that represents a collaboration

Figure 1 is an example of SCG in which we use solid directed edges to represent logic dependencies as opposed to the dotted edge to represent data dependencies. As an example for data dependency, W_4 data-depends on W_1.

3.2. Trust model

Trust is a kind of relation that describes how much one believes in another. Evaluating trust for service can help users to predict its future behavior [7]. For service collaboration, we denote its trust value as $T_c(x)$. In our trust model, we consider three kinds of elements to derive the final trust value of a service collaboration: the trust for individual partner services, the explicit trust relation among partner services that have logical dependencies for each other, and the implicit trust relation implied in data-dependencies among services.

3.2.1. Trust for single partner service

How to evaluate the trust of partner service is not the focus of this paper. At present, there has been much work on it. So we adopt the trust mechanism proposed by Malik and Bouguettaya [18] to evaluate the trust of partner services because the mechanism takes consideration of multiple QoS aspects of services, credibility of raters, preference of consumers, time sensitivity, etc., and is proved to be comprehensive and effective for service trust. According to the trust assessment, given a service s_0, the trust value T_0 for s_0 is calculated as follows:

$$T_S(s_0) = \frac{\sum_{c=1}^{L} PerEval_{s_0}^{c} \cdot fd(t) \cdot C_r(c)}{\sum_{c=1}^{L} C_r(c)} \quad (1)$$

L is the set of service consumers that have invoked s_0. $PerEval^c_{s_0}$ is the personal evaluation value given by the consumer c. $C_r(c)$ is the creditability of c. $fd(t)$ is a function that makes the evaluation fade with time.

3.2.2. Trust for logic dependency

The relations of logic dependency among partner services decide the execution order and process of service collaboration. As mentioned in section 3.1, there are multiple types of logic dependencies. It is intuitive that developing different trust assessment algorithm for different types of logic dependencies respectively may cause high time complexity when the structure of collaboration becomes complicated. So we use a path based method to evaluate trust for logic dependency. For service collaboration, there is usually a start service getting the initial requests from a customer and an end service that returns the final results to the customer. There may be several paths from the start service to the end service. As each path is possible to be executed, we can distribute a weight to each execution path based on the frequency of execution for each path according to the historic execution:

$$T_C(x) = \sum_{i=1}^{n} T_{p_i} \cdot \lambda_i \tag{2}$$

n is the number of execution paths. T_{p_i} is the trust value for the execution path p_i. λ_i is the frequency of execution for p_i. If the collaboration is executed for the first time, the weights of each path will be initialized by the consumer.

3.2.3. Trust for data dependency

In order to numerically estimate the trust relation between two data-intensive services with data-dependency, which means the extent to which a service a trusts the data from b, we import the concept Trust Degree denoted as $TD(a,b)$ which can be calculated according to the past experience of invocation on b initiated by a. The computing formula is as follows.

$$TD(a,b) = \frac{\sum_{i=1}^{n} PerEval^a_b \cdot fd(t_i)}{n} \tag{3}$$

n is the number of times a invoked b in the past. $PerEval^a_b$ is the personal evaluation value given by a. $fd(t_i)$ is the same as the function aforementioned. A service can easily get the Trust Degree in another with which it has interacted before according to the above formula. But for the services that have never interacted with directly before, it can get Trust Degree through the following method based on the characteristic of trust transitivity.

Trust is strictly conditional transitive [19]. In general, if a trusts b with the degree x and b trusts c with the degree y, then the degree of a's trust in c is associated with x and y, and can be denoted as $u = f(x,y)$. Trust is not always transitive. Only under certain semantic constraints can trust be transitive [20]. For example, the fact that a trusts b to repair his car and b trusts c as a good wine merchant, does not imply that a also trusts c to repair his car. We assume that the participants in service collaboration are all under a certain semantic constraint, so that trust can be transitive among them.

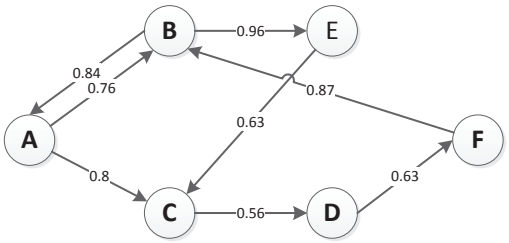


Figure 2: A weighted digraph that represents a trustworthy network

Trust relations between services can construct a trustworthy network. It can be transformed to a weighted digraph $G = (V,E,\tau)$[21], where V is the set of services that have historic interaction experience. E is the set of edges which show the direct trusts. τ is the weight function which is equal to Formula (3). The weight of an edge (m,n) is the Trust Degree of node m over n. Thus, the problem of measuring Trust Degree of two services never interacting with each other before can be transformed to the problem of getting the resultant weight between two nodes that there's no direct edge between them (just like node A and F in Figure 2).

Definition 4 (Transitive Trust Degree). Given two services a and b which never interacted with each other before. But there are several trust paths between them. The derived trust degree is called Transitive Trust Degree, denoted as $TD(a:b)$.The symbol ':' means that the two services have never interacted with each other before. The derivation principles are as follows:

(1) For each path p_i from a to b, we use probability product to derive trust degree of a in b through path p_i. Because trust evaluation from different services is made independently and the final evaluation is affected by all the transitive evaluation. This can also ensure the decrease of transitive trust degrees with the increment of the length of the trust path:

$$TD(a:b,p_i) = \prod_{i=1}^{n-1} TD(k_i,k_{i+1}) \tag{4}$$

n is the number of nodes on the path p_i. k_i is a certain node on the path. k_1 is a and k_n is b.

(2) There may exist several parallel paths from a to b, among the values of the derived trust degree through these paths, we choose the maximum value as the Transitive Trust Degree of a in b:

$$TD(a:b) = \max\{TD(a:b,p_i) | p_i \in P\} \quad (5)$$

Let's consider some cases below. There are two paths from a to b,($a \to c \to b$ and $(a \to d \to b)$:

(a) $TD(a,c) = 0.9, TD(c,b) = 0.9$,
$TD(a,d) = 0.1, TD(d,b) = 0.1$

It's intuitive that the trust degree of a in b is close to 0.9, because a trusts c and distrusts d and thus a can neglect the opinion of d. With the proposed method, we get $TD(a,c) = 0.81$, which means a trusts b.

(b) $TD(a,c) = 0.1, TD(c,b) = 0.9$,
$TD(a,d) = 0.1, TD(d,b) = 0.1$

a distrusts both c and d, so the opinions of c and d can't be trusted by a no matter b is recommended or not. With the proposed method, we get $TD(a,c) = 0.09$, which means a distrusts b.

(c) $TD(a,c) = 0.9, TD(c,b) = 0.9$,
$TD(a,d) = 0.9, TD(d,b) = 0.1$

a trusts c as well as d, but c and d have opposite opinions on b. Here our method is optimistic, we choose the positive opinion. With the proposed method, we can get $TD(a,c) = 0.81$, which means a trusts b.

(d) $TD(a,c) = 0.9, TD(c,b) = 0.1$,
$TD(a,d) = 0.1, TD(d,b) = 0.9$

a trusts c but c distrusts b, a distrusts d but d trusts b. Through both of the paths, we can conclude that a won't trust b. With the proposed method, we get $TD(a,c) = 0.09$, which means a distrusts b.

4. Trust Evaluation Process

In this section, we present the process of evaluating the trust for service collaboration of data-intensive services with the help of the results of Section 3. It consists of two main steps. The first is to assess the trust of each path from the start service to the end one with the consideration of data-dependency. The second step is to evaluate the trust for the whole collaboration by means of summarizing the trust for each path.

4.1. Evaluate Trust for Paths

For a SCG, we use a sequence $< s_1, ...s_i, ...s_n >$ to represent a logic path from the start service to the end service along logic dependencies; and each element is a partner service on the path. Note that different partner services on the path may differ in the contribution of the trust evaluation for the path. We import the concept of Critical Service to represent the most important partner services on a path and define it as follows.

Definition 5 (Critical Service). For a logic path p of a SCG and a service s on p, s is a Critical Service on p if and only if the condition $D(s) = \max(D(s_i))(1 \le i \le n)$ holds where the function $D(s)$ is to get the number of data dependencies emitted from the node s in the SCG.

Critical Service is such a service node in a SCG that it is depended on by other service nodes to the maximum extent. Thus, the above definition shows that a Critical Service of a path has the biggest out degree of data dependency on this path. Note that there may be more than one critical service on a path. In Figure 1, for the path $< W_1, W_2, W_4, W_5 >$, W_1 and W_4 are both the critical services on this path. If a critical service breaks down, then the partner services that data-depend on it cannot execute because of the lack of inputs. In this case, the path will fail to execute. So the critical service is more important than the other partner services in the path and it should be assigned with a heavier weight than others.

In order to compute the trust value for each execution path, we combine the trust assessment for partner services and the trust relation implied in data-dependencies between partner services:

$$T_p = \frac{\omega_1 \cdot \sum_{i=1}^{m_1} T_{S_i} + \omega_2 \cdot \sum_{j=1}^{m_2} T_{S_j} + \omega_3 \cdot \sum_{k=1}^{n} T_{D_k}}{m_1 + m_2 + n} \quad (6)$$

T_{S_i} is the trust value of the critical services. T_{S_j} is the trust value of the other partner services on this path. T_{D_k} is the value of data-dependency trust which can be computed according to section 3.2. m_1 and m_2 are the numbers of critical services and normal services. n is the occurrence number of data -dependencies between partner services. The parameters ω_1, ω_2 and ω_3 are the weights of the above values. As mentioned before, the critical services are more important, so ω_1 is set to be bigger than the other two.

For each logic path, the algorithm TELP (Trust Evaluation for Logic Path) is designed to evaluate the trust for the path according to Formula (6). It firstly evaluates all the partner services on this path and finds the critical services. Then it finds the dotted edges on this path, and calculates the value of TD between the two nodes of each dotted edge.

4.2. Evaluate Trust for Service Collaboration

After getting the trust for each logic path in service collaboration, we can evaluate the trust for the whole collaboration according to Formula (2). In order to do that, we first find all the logic paths in the service collaboration. This can be done with the help of the following algorithm which is based on the idea of Depth-First Traversal. The time complexity of the algorithm is $O(e \cdot v)$ where e is the number of edges and v is the number of vertexes.

5. Experiments

In order to check the accuracy of the proposed trust evaluation method for service collaboration, we conduct the experiment with the help of the simulation software

Algorithm 1 TELP (Trust Evaluation for Logic Path)

Require:
P - A sequence that represents an execution path;
ω_1 - Weights of critical services;
ω_2 - Weights of other partner services;
ω_3 - Weights of Trust Degree between two partner services;
Ensure:
T_P - Trust Degree of the path P;
1: Initialize $T_p = 0$, n is the number of partner services, m is the number of data dependencies;
2: **for** each partner service s in P **do**
3: calculate the trust value of s, marked as $T_s(s)$;
4: **if** s is a critical service **then**
5: $T_p += \omega_1 \cdot T_S(s)$;
6: **else**
7: $T_p += \omega_2 \cdot T_S(s)$;
8: **end if**
9: **end for**
10: **for** $i = 0$ to n **do**
11: **for** $j = 0$ to n **do**
12: **if** s_i data-depends on s_j **then**
13: calculate $TD(s_i, s_j)$;
14: $T_p += \omega_3 \cdot TD(s_i, s_j)$;
15: **end if**
16: **end for**
17: **end for**
18: $T_p = T_p/(n+m)$;
19: **return** T_p;

Algorithm 2 FindAllpath

Require:
G - a Service Collaboration Graph;
Ensure:
All the logic paths of G;
1: Initialize start = the start node of G, end = the end node of G;
2: path = null; (an array of nodes)
3: GetPaths(G, path, start, end, 1);
4: GetPaths(G, path, startnode, endnode, length)
5: **if** startnode has been visited **then**
6: **return** ;
7: **end if**
8: **if** startnode == endnode **then**
9: output the path;
10: **else**
11: set startnode to be visited;
12: **for** $i = 0$ to total nodes of G **do**
13: **if** there is path from startnoed to i and i has not been visited **then**
14: GetPaths(G, path, i, endnode, length + 1);
15: **end if**
16: set i to be unvisited;
17: **end for**
18: **end if**

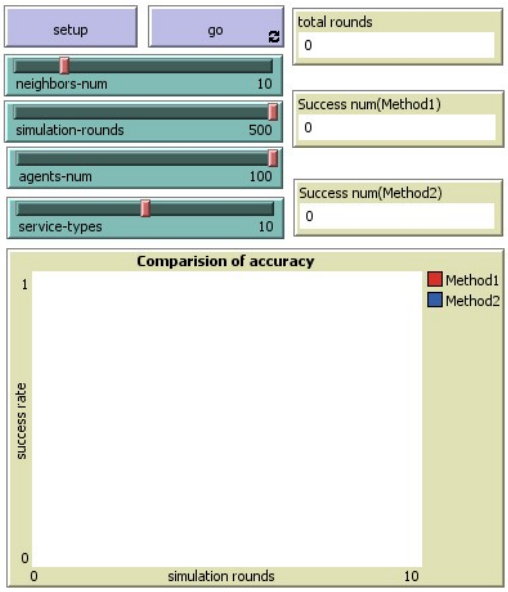


Figure 3: Simulation application

NetLogo. The software NetLogo is a multi-agent programmable modeling environment, which can model, manipulate and inspect agents [6].

Table 1: Simulation parameters

Parameter	Value
Number of agents	100
Number of service types	10
Max number of neighbors	10
Total simulation rounds	500

5.1. Experiment Setup

The experiment is carried out under the environment of Windows 7, NetLogo 4.1.2, Intel Core2 2.4G CPU and 4G RAM. In the simulated trust community, 100 agents are generated with a random trust value to provide and consume services. There are 10 types of services in all and each agent has some services belonging to the 10 types randomly. The trust value of one service equals to the value of its owner agent. We define neighbors of one agent as the agents that it has interacted before, thus the trust degree between them can be produced. The max number of one agent's neighbors is 10. The values of trust degree between agents with their neighbors are also generated randomly. The simulation parameters are listed in Table 1.

The simulation application we developed with NetLogo is shown in Figure 3. It supports adjusting the

simulation parameters using several scroll bars. The button SETUP is to initialize the simulation world according to the simulation parameters. The button GO is to start the simulation. The four slides map to the four parameters. Three monitor windows are on the top-right, which monitor the statistics. The chart at the bottom will show the simulation result. The tool can be downloaded from http://www.rayfile .com/files/e830f0c0-35a3-11e0-a11b-0015c55db73d/.

5.2. Experiment Results and Analysis

We conduct 500 rounds of simulation totally. For each round, we issue a collaboration request randomly, and make the agents to participate in collaboration to satisfy the request in different methods. Since we focus on the trust evaluation, we omit the functional selection of services. So the request is an execution flow of different types of services and we assume each service data-depends on the pre-service in the flow. Then we use different methods to select trustworthy agents that provide trustworthy services to participate in the collaboration. The first method is from [17], which evaluates trust for collaboration considering trust value of each component and the collaboration structure. In another word, if the structure is determined, it will choose the most trustworthy service for each node in the execution flow. The second method is our proposed method. We select a trustworthy agent not only considering its trust value but also the trust degree in the agents on which it data-depends. The trust degree between two agents can be computed according to section 3.2. In order to verify the accuracy of the two methods, we respectively simulate the interaction of the collaboration with the partner services chosen by each method. For each agent, there is a trust threshold to trust others, if the value of trust degree is under the trust threshold, the agent will distrust another. Here we set the trust threshold as 0.5. That means, if the trust degree between one agent with another agent on which it data-depends is less than 0.5, the interaction between them will fail. Then the agents in the collaboration cannot cooperate successfully. The accuracy of each method is equal to the success rate of the cooperation. Finally we can compare the accuracy of different method.

The simulation result is shown in Figure 4. During the 500 rounds of simulation, the success number of cooperation among agents chosen with the first method is 252, which is less than that of our method 439. The accuracy of the two methods is 50.4% and 87.7%, respectively. Besides, we can also see that the convergence rate of our method is also higher than the other. So we can conclude that it's quite significant to take consideration of the trust relationship between components with data-dependency when evaluating trust for service collaboration.

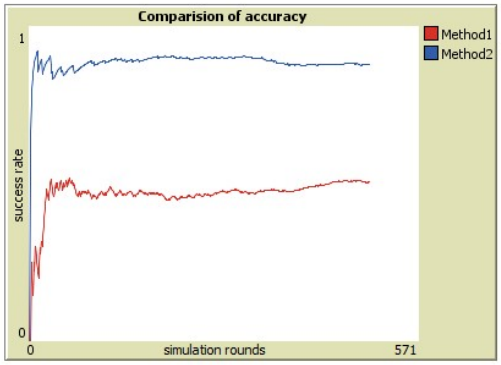


Figure 4: Comparison of accuracy between two methods(Red line represents the accuracy of method1,blue line represents the accuracy of method2)

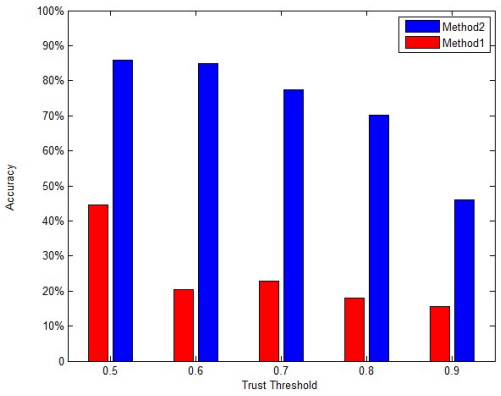


Figure 5: Accuracy of the two methods under different trust thresholds

Since trust is subjective, different agent may suffer different trust value. In the above experiment, we set the trust threshold as 0.5. In order to check the performance of the methods under different trust threshold, we make the simulation with the value of trust threshold from 0.5 to 0.9. The result is shown in Figure 5, which indicates that our method can reflect people's subjective opinion better.

6. Conclusion

This paper introduces a formal trust model for service collaboration of data-intensive services with the consideration of data-dependency. And it also proposes a numerical estimation method based on the characteristic of trust transitivity to assess the trust relation between two services that have never interacted with each other before. In order to evaluate the trust for the whole service collaboration, we find out all the execution paths first and

then evaluate trust for each path with data dependencies. Finally, we aggregate the trust for each path to get the collaboration trust. A serial of experiments using NetLogo show that the method can polish up the accuracy of trust evaluation.

Trust evaluation is a very important issue for service collaboration in the cloud environment. In the future, we will improve the proposed method by considering more complicated structure in service collaboration, for example the none-well-defined structure. And also, we are considering the trust evaluation for service collaboration with the hybrid participants of data-intensive services and others.

Acknowledgement

This research was supported by the the National Key Technology R&D Program 2011BAD21B02, National Natural Science Foundation of China under Grant No. 61170033 and Zhejiang Provincial Natural Science Foundation of China under Grant No.Y1080372.

References

[1] Feng DG, Zhang M, Zhang Y, Xu Z; Study on cloud computing security. Journal of Software, 2011, **22(1)**: 7183.

[2] Elliot K. Kolodner; Data-intensive Storage Services on Clouds: Limitations, Challenges and Enablers; 2nd Workshop on Software Services: Cloud Computing and Applications based on Software Services, 2011.

[3] Kerschbaum Florian, Haller Jochen, Robinson, Philip; PathTrust: A Trust-Based Reputation Service for Virtual Organization Formation; Trust Management **3986**: 193-205, 2006.

[4] Schubert Lutz, Tuptuk Nilufer, Telecom British; The TrustCoM Approach to Enforcing Agreements between Interoperating Enterprises; Enterprise Interoperability Part VII: 365-375, 2007.

[5] Touhid Bhuiyan , Audun Jsang, Yue Xu; An Analysis of Trust Transitivity Taking Base Rate into Account; In: proceeding of the Sixth International Conference on Ubiquitous Intelligence and Computing, 2009.

[6] NetLogo, http://ccl.northwestern.edu/netlogo/.

[7] Audun Jsang, Roslan Ismail, Colin Boyd; A survey of trust and reputation systems for online service provision; Decision Support Systems **43(2)**: 618-664, 2007.

[8] Yao Wang and Julita Vassileva; A Review on Trust and Reputation for Web Service Selection; In: Proceeding of the 1st Int. Workshop on Trust and Reputation Management in Massively Distributed Computing Systems, Toronto, Canada, June 2007.

[9] Hien Trang Nguyen, Weiliang Zhao, Jian Yang; A Trust and Reputation Model Based on Bayesian Network for Web Services; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2010.

[10] Zaki Malik, Ihsan Akbar and Athman Bouguettaya; Web Services Reputation Assessment Using a Hidden Markov Mode; In: Proceedings of the 7th International Conference on Service Oriented Computing, 2009.

[11] Surya Nepal, Wanita Sherchan, Jonathon Hunklinger, etc; A Fuzzy Trust Management Framework for Service Web; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2010.

[12] Haihua Li, Xiaoyong Du and Xuan Tian; A Capability Enhanced Trust Evaluation Model for Web Services; Chinese Journal of Computers **31(8)**: 1471-1477. 2008.

[13] Weiliang Zhao and Vijay Varadharajan; Trust Management for Web Services; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2008.

[14] Damjan Kovac and Denis Trcek; Qualitative trust modeling in SOA; Journal of Systems Architecture **55(4)**: 255263, 2009.

[15] Surya Nepal, Zaki Malik and Athman Bouguettaya; Reputation Propagation in Composite Services; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2009.

[16] Hamdi Yahyaoui; Trust Assessment for Web Services Collaboration; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2010.

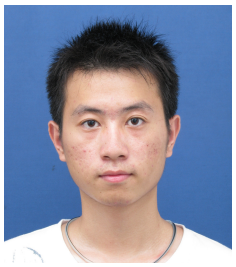
[17] Sharon Paradesi, Prashant Doshi and Sonu Swaika; Integrating Behavioral Trust in Web Service Compositions; In: Proceedings of the IEEE International Conference on Web Services, ICWS, 2009.

[18] Zaki Malik, Athman Bouguettaya; RATEWeb: Reputation Assessment for Trust Establishment among Web services; The VLDB Journal **18(4)**: 885911, 2009.

[19] Xiaoqing Zheng, Huajun Chen, Zhaohui Wu and Yu Zhang; A Computational Trust Model for Semantic Web Based on Bayesian Decision Theory; Lecture Notes in Computer Science **3841**: 740-745, 2006.

[20] Audun Jsang and Simon Pope; Semantic Constraints for Trust Transitivity; In: Proceedings of the Asia-Pacific Conference of Conceptual Modeling (APCCM), Newcastle, Australia, February, 2005.

[21] Yixiang Chen, Tian-Ming Bu, Min Zhang, etc; Measurement of Trust Transitivity in Trustworthy Networks; Journal of Emerging Technologies in Web Intelligence, **2(4)**: 319-325, 2010



Longtao Huang received the Bachelor's degree in software engineering from Zhejiang University, Hangzhou, China, in 2010. Now study the PhD of computer science and technology in Zhejiang University from 2010. His research interests include service computing and cloud computing.

Shuiguang Deng received the Doctor's degree in computer science from Zhejiang University, Hangzhou, China, in 2007. He is currently an associate professor in Zhejiang University. His research interests include service computing, cloud computing and middleware techniques.

Ying Li received the Doctor's degree in software technology from Zhejiang University, Hangzhou, China, in 2000. He is currently an associate professor in Zhejiang University. His research interests include software architecture, software automation, compiling technology, and middleware techniques.

Jian Wu received the Doctor's degree in computer science from Zhejiang University, Hangzhou, China, in 2004. He is currently an associate professor in Zhejiang University. His research interests include service computing, middleware techniques, data mining, and intelligent transportation.

Jianwei Yin received the Doctor's degree in computer science from Zhejiang University, Hangzhou, China, in 2001. He is currently a professor in Zhejiang University. His research interests include distributed network middleware, software architecture, information integration, and information security.

Gexin Li received the Master's degree in physics from South-Central University For Nationalities, Wuhan, China, in 1993. He is currently an associate professor in Whenzhou University. His research interests include computer network, web information engineering.