Journal of Computers Vol. 28, No. 6, 2017, pp. 236-245
doi:10.3966/199115992017122806021

MCTModel: A Multi-clouds Trust Model Based on SLA in Cloud Computing

Zhen-Hua Tan[1*], Yi-Cong Liu[1], Nan-Xiang Shi[1], Xing-Wei Wang[1] and Nan Guo[2]

[1] Software College, Northeastern University,
Shenyang, Liaoning, China
tanzh@mail.neu.edu.cn, 1571488@stu.neu.edu.cn,
1571491@stu.neu.edu.cn, wangxw@mail.neu.edu.cn

[2] Faculty of Computer Science and Engineering, Northeastern University,
Shenyang, Liaoning, China
guonan@mail.neu.edu.cn

Received 27 March 2017; Revised 29 June 2017; Accepted 03 July 2017

Abstract. Trust has become a critical issue in the development of cloud computing in recent years. In most of the existing models, only a single cloud provider (called single cloud) is selected to trade with the cloud consumers. In this paper, in order to select reliable cloud providers, we proposes a novel trust model based on service level agreement (SLA) using multiple clouds, named MCTModel. In the model, a cloud consumer can apply for the multiple service providers for their own services, and trust value is updated according to the cloud providers' performance of SLA in the process of service usage for the different demands for the different attributes. The time decay model is applied to calculate the value of trust, making the trust value more accurate. Simulations show trust model of the multiple clouds can ensure good service to users compared the trust model of single cloud.

Keywords: cloud computing, multiple clouds, SLA, trust model

1 Introduction

Cloud computing has been a hot topic in recent years, and it is a business computing model where the computing tasks are distributed in the resource pools consisting of a large number of computers. Users are allowed to access the resources, which can be quickly supplied and distributed, like computing power, storage space and information services in a pay-and-use way. Cloud computing has the following characteristics: ultra-large-scale, on-demand service, ubiquitous network access, rapid elasticity, cheap cost. In cloud computing, cloud service providers (referred CSP) provide the consumers with large-scale resources as service to the cloud service consumers (referred CSC) and consumers pay it on demand.

However, with the development of cloud computing, many problems have emerged, among which trust is the most prominent problem. As the quality of service provided by different CSPs varies, the lack of adequate knowledge about the quality of CSP services and concerns about product security led to a lack of trust for the CSP. How to choose the most trusted CSP to meet the needs of CSCs and build the trust model become a hot topic.

Trust is a subjective concept. To build a trust model in cloud computing needs to consider the following problems. (1) How to reduce the effect of CSCs' subjective evaluation on trust value; (2) How to prevent the collusion case of CSC; (3) How to prevent cloud providers from providing malicious fake service; (4) Trust is multifactorial, How to make a more comprehensive excavation on trust related factors, for instance, the success rate of the transaction and so on for the assessment of trust makes the assessment results more reasonable; (5) Incentive measures, how to motivate the reliable entities and

* Corresponding Author

punish the malicious entities in the trust system are essential and so on.

The existing trust model still has some problems. (1) most trust models are single cloud. In the process of the transaction, the cloud consumer chooses a cloud provider to carry on the transaction, and cannot change the providers if the provider provides the bad service; (2) There are many factors that need to be considered in building trust model, trust parameters are not fully mined to be used for trust modeling in some models; (3) Some models do not consider the case of collusion. (4) Trust has the property of time decay. As time goes on, if the provider doesn't trade with cloud consumers for a long time, then the trust value should be reduced. Some trust models do not consider the impact of time factor on trust value.

In this paper, a multiple clouds trust model based on SLA, named MCTModel in short, in which the users' subjective evaluation is abandoned to reduce the effect of subjective scores on trust, is proposed in order to select trusted service providers. Meanwhile, in this model, a cloud consumer can use the services provided by multiple cloud providers, which reduces the impact of single provider failing to well perform services to the cloud consumers.

2 Related Work

Domestic and foreign experts have proposed a series of trust models for different areas of research using different methods and tools.

Alhamad, Dillon and Chang [1] presented the main criteria of SLA, defined dynamic SLA metrics for different groups of cloud users and investigated the negotiation strategies between cloud providers and cloud consumers in the cloud computing. However, the paper did not design the SLA metrics specifically and implement the simulation process for the framework proposed. After that, aiming at helping cloud consumers choose the most reliable resources, Alhamad, Dillon and Chang [2] proposed a trust evaluation model based on SLA to evaluate cloud services, and a novel trust structure using SLA and business monitoring activities to ensure the quality of cloud services. While this paper only presented a conceptual description, it did not give a specific evaluation process as well as how to use the monitoring results.

Alhamad, Dillon and Chang [3] examined the some related challenges about the concepts of trust, SLA management and cloud computing, and then discussed the existing framework of SLA in different areas and advantages and limitations of performance measurement model. Saleh, Hamed and Hashem [4] proposed a hybrid model to build, evaluate, and expose trust for ensuring the credibility of the entities and a model architecture of service-oriented that treated trust as a service to delivery and a module for service registry and novel personalized modules were added for tracking both the behaviors of the cloud providers and cloud consumers, effectively blocking the malicious behaviors of cloud users. But the model did not consider collusion.

Li and Du [5] introduced an adaptive trust management model to effectively evaluate the competence of cloud services based on the multiple trust attributes and the two kinds of adaptive modeling tools (rough sets and induced ordered weighted averaging operator) were organically combined to the data mining and knowledge discovery. To cope with the strategically altering behavior of malicious agents and distributing workload as evenly as possible among service providers, Das and Islam [6] put forward a dynamic trust model based on feedback and a new load balancing algorithm.

Wang et al. [7] presented a platform of Service Level Agreement, and then a reputation system was proposed based on the platform to assess the reliability of the provider. Meanwhile, a SLA template pool was proposed to make the SLA negotiation more convenient between cloud consumers and cloud providers. But there were no specific mathematical algorithms and simulation experiments. Manuel [8] used the past credentials and present capabilities of a cloud resource provider to construct a novel trust model and calculated the value of four related parameters.

Mohsenzadeh and Motameni [9] considered success and failure interactions between cloud entities to introduce a trust model based on fuzzy mathematics in cloud computing environment. Noor, Sheng, Maamar and Zeadally [10] described different trust management perspectives and techniques, proposed a generic analytical framework with a set of 14 criteria to assess trust management systems in cloud computing, and discussed open research challenges revealed by an analysis of 30 available systems. Jules, Hafid and Serhani [11] proposed a framework to choose a trusted provider based on its reputation for cloud consumers, and a dynamic SLA scheme using a probabilistic ontology capable of detecting potential violations of contract parameters.

Tan, Wang, Cheng, Chang and Zhu [12] presented a new distributed trust model. They designed a communication multi-dimension history vector and its distributed storage structure, taking into account a number of factors to calculate the value of trust. Tan, Wang and Wang [13] proposed a dynamic and iterative trust model, and the latest evidence was added to the iterative calculations. Wang and Vassileva [14] proposed a trust model based on Bayesian network integrating Bayesian concepts into the trust model, but did not give the process of calculation.

Saini, Sihag and Yadav [15] reviewed existing collusion attacks, proposed a reactive defense mechanism against such collusion attacks, and provided a reduction mechanism to chastise colluded peers. Binu and Gangadhar [16] proposed a SLA framework consisting of negotiation and secure monitoring mechanism and a third party is developed. Li, Wang, Kang, Guo and Cao [17] established a framework of trust evaluation system and proposed a trust evaluation model oriented to mechanical manufacturing field.

In [18], five parameters (availability, reliability, data integrity, identity and capability) are used to evaluate the trust value by considering the influence of opinion leaders on other entities and removing the troll entities effect in the cloud environment. Also, they proposed a method for opinion leaders and troll entity identification. Noor, Sheng, Yao, Dustdar and Ngu [19] described the design and implementation of a reputation-based trust management framework.

Many literatures discuss on trust problem. However, due to limitations of space, we are unable to present all the existing body of literature. In cloud computing, the execution of services has changed to be completely independent of the consumers' infrastructure. So, cloud computing needs dynamic mechanism. Our proposed model will present a novel trust model for cloud computing.

3 Algorithm Overview of MCTModel

In MCTModel, cloud computing includes two kinds of entities, cloud providers and cloud consumers. When the cloud consumers need cloud providers to provide services according to their own needs, cloud providers are chosen from set of cloud providers available, and then the two sides begin to negotiate the details of transaction and sign the service level agreement. After the end of the transaction, the trust value of cloud providers is updated. The detail procedure of algorithm proposed is shown in Fig. 1.

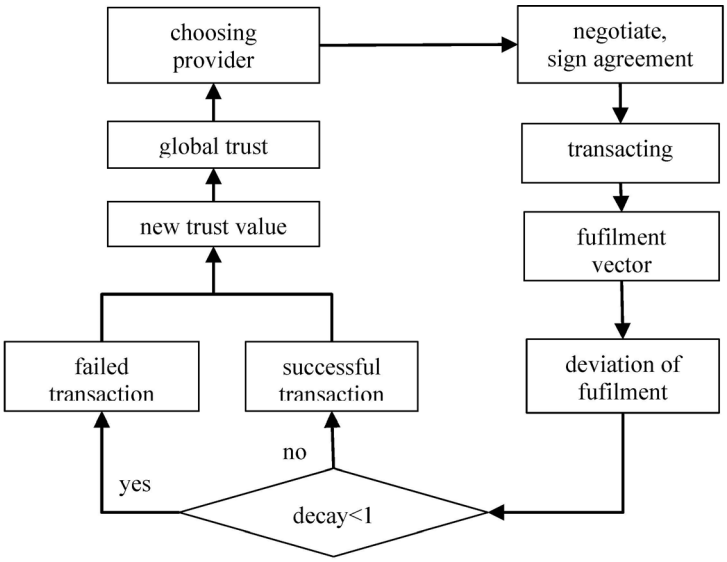

Fig. 1. Algorithm overview

Step 1. First of all, the trusted providers are selected to meet the needs of cloud consumers according to trust value of the provider's before the start of the service;
Step 2. Cloud providers negotiate with cloud consumers, and sign SLA agreement;
Step 3. Starting trading, and then gain fulfillment vectors;
Step 4. According to the needs of cloud consumers and the completion of the parameter vectors to make SLA judgment, and to see whether the cloud provider well performed SLA protocol content;

Step 5. If cloud service provider does not perform properly, the transaction is recorded a failed one, trust value will decline; in the contrast, trust value will increase;
Step 6. Consider the time factor, and calculate the global trust value.

4 Proposed MCTModel

4.1 Trust Parameters in MCTModel

In this section, we discussed several parameters related to the quality of service to measure the pros and cons of cloud services in MCTModel, helping consumers choose more credible providers. From the perspective of SLA and quality assurance, three kinds of trusted factors are considered mainly, integrity, availability and reliability.

Integrity. Integrity, here we are referring to the integrity of the tasks. The worse the situation of network is, the easier to lose the tasks are. It uses the task loss rate (referred to *TL*) to represent the integrity, as defined formula (1) below.

$$TL_{Pr}^{n} = \frac{N_r^{loss}}{N_r^{tot}} \tag{1}$$

TL_{Pr}^{n} represents the task loss rate of the *n*-th transaction for the cloud providers *r*. N_r^{loss} represents the number of lost tasks in the *n*-th transaction, N_r^{tot} represents the total number of tasks in the *n*-th transaction for the cloud providers *r*.

Availability. Availability means the service resources and data storage within a cloud which is made accessible and usable on demand by an authorized entity. Availability is measured in terms of average response time (referred to *RES*) and the definition of *RES* is followed as formula (2).

$$RES_{Pr}^{n} = \frac{\sum_{i=1}^{m} RES_r^i}{m} \tag{2}$$

RES_{Pr}^{n} expresses the average response time of the *n*-th transaction in which the tasks have been accepted and made successful responses in a single transaction for provider *r*. RES_r^i expresses the response time of *i*-th task in the *n*-th transaction, and is equal to the difference of receiving time and the response time. *m* is the total number of tasks sent and responded successfully.

Reliability. Reliability is an important parameter. It uses the task success rate (*SU*) to represent (if cloud consumers in the task list, submit a total of ten tasks, and providers receive only seven, but only five are completed successfully, the task success rate is 0.5), as defined formula (3) below.

$$SU_{Pr}^{n} = \frac{N_n^{suc}}{N_n^{tot}} \tag{3}$$

SU_{Pr}^{n} represents the task success rate of the *n*-th transaction in which the tasks are sent for cloud provider *r*. N_n^{suc} represents the number of the tasks requested and completed successfully in the *n*-th transaction. N_n^{tot} represents the total number of tasks in the *n*-th transaction for the cloud providers *r*.

As dimensions of different SLA parameters are different, and some elements are positive indicators, the bigger the better, such as the successful rate; some elements are reverse indicators, the smaller the better, such as the average response time, and task loss rate. Parameters monitored need to be normalized in advance, which are converted to (0,1). The process is as follows:

$$SU_{Pr}^{n} = \frac{SU_{Pr}^{n} - \min(SU_{Pr}^{n})}{\max(SU_{Pr}^{n}) - \min(SU_{Pr}^{n})} \tag{4}$$

$$TL_{Pr}^{n} = \frac{\max(TL_{Pr}^{n}) - TL_{Pr}^{n}}{\max(TL_{Pr}^{n}) - \min(TL_{Pr}^{n})} \tag{5}$$

$$RES_{Pr}^{n} = \frac{\max(RES_{Pr}^{n}) - RES_{Pr}^{n}}{\max(RES_{Pr}^{n}) - \min(RES_{Pr}^{n})} \tag{6}$$

Here $\max(SU_{Pr}^{n})$ and $\min(RES_{Pr}^{n})$ respectively represent the maximum and minimum value of SU_{Pr}^{n}, so do the $\max(TL_{Pr}^{n})$, $\min(TL_{Pr}^{n})$, $\max(RES_{Pr}^{n})$ and $\min(RES_{Pr}^{n})$.

4.2 Trust Evaluation in MCTModel

SLA trust (referred *sT*) is the trust value formed after the transaction. SLA trust is updated based on the degree of fulfillment of SLA parameters during the service process of the SLA parameters. sT_{r}^{n} represents the trust value of providers' *n*-th transaction between the cloud providers and cloud consumers for cloud provider *r*.

Deviation of fulfillment. Considering the different needs of different users for different attributes, the need of users needs to be considered for different attributes for choosing cloud providers. $W_r = (w_r^1, w_r^2, \ldots, w_r^3)$ represents the degree of user demand for different attributes, and $w_r^1 + w_r^2 + w_r^3 = 1$.

The fulfillment vector of SLA at the consumer side is $V_{Pr}^{n} = (TL_{Pr}^{n}, RES_{Pr}^{n}, SU_{Pr}^{n})$ after the processing above while the SLA contract vector is $V_{Pr}^{c} = (TL_{Pr}^{c}, RES_{Pr}^{c}, SU_{Pr}^{c})$ which is consulted by the cloud providers and the cloud consumers before the *r*-th transaction. *DF* represents the deviation of fulfillment between V_{Pr}^{n} and V_{Pr}^{c}, as shown in (7).

$$DF_{Pr}^{n} = W_r * (V_{Pr}^{n} - V_{Pr}^{c}) \tag{7}$$

The value of *DF* may be two cases, positive or negative. The positive represents that the fulfillment of the service is higher than the demands of cloud consumers under the account of the needs of the cloud consumers, and the cloud providers well perform the services of cloud consumers; the negative represents that the fulfillment of the service is lower than the needs of cloud consumers, cloud providers badly perform the needs of cloud consumers. *d* is used to represent the degree of decay of fulfillment. The expression of *d* is expressed as an exponential function.

$$d = \begin{cases} 1 & \text{if } DF_{Pr}^{n} > 0 \\ \gamma^{DF_{Pr}^{n}} & \text{if } DF_{Pr}^{n} < 0 \end{cases} \tag{8}$$

Update of trust. The trust values of cloud consumers are updated using the method of iteration according to the fulfillment of the service during the process of transaction.

If the result of formula (8) exceeds the allowed range τ, the abnormal result is informed of the cloud consumers, thus cloud consumers will no longer send tasks or requests to the providers. The transaction is recorded as a failed transaction and cloud consumers stop the service providers to continue to provide services. The SLA trust value of the service providers will decline, as the formula (9).

$$sT_{r}^{n} = \begin{cases} 0 & , \text{ if } DF_{Pr}^{n} < 0 \text{ and } sT < 0 \\ sT_{r-1}^{n} - \dfrac{1 - d * sT_{r-1}^{n}}{\varphi} & , \text{ if } DF_{Pr}^{n} < 0 \text{ and } sT > 0 \end{cases} \tag{9}$$

In this case, the trust value can't be negative. The minimum is 0.

If the result of the formula (8) is within the allowed range τ, that is to say, the service providers are providing normal service and continue to provide service until the result of *d* exceeds the allowed range or the transaction completes. Completion of transaction represents a successful transaction, and the service providers need be encouraged to increase their dynamic trust values, as the formula (10).

$$sT_{r}^{n} = sT_{r-1}^{n} + \frac{1 - d * sT_{r-1}^{n}}{\mu} (DF_{Pr}^{n} > 0) \tag{10}$$

Wherein $\mu \geq \varphi$, because the intensity of punishment must be greater than the intensity of reward.

Time factor. Due to the dynamic characteristics of cloud environment, with the time going by the trust will weaken in the absence of the transaction. If cloud providers have been idle for a long time, the trust value will decline. Using exponential function calculates the decay factor, which is used to calculate global trust. The time decay function follows the principle that the closer the transaction is, the more reliable the transaction is and is defined as formula (11) below.

$$\rho = e^{-\beta * t} \tag{11}$$

where $t = t_{current} - t_{previous}$, and β is the decay rate.

The time decay curves of different parameters are shown in Fig. 2

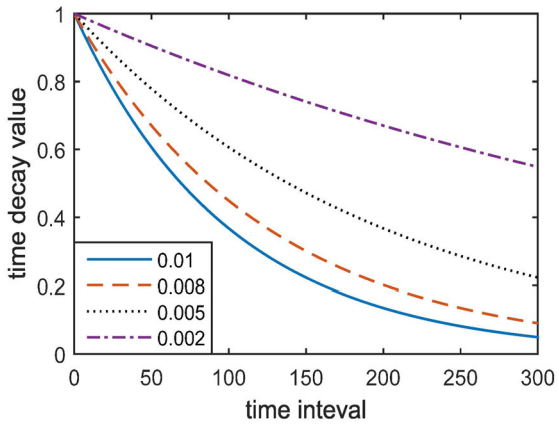

Fig. 2. Time decay curves

4.3 Global Trust Computing Algorithm in MCTModel

Now, the global trust gT_r^n represents the global trust value of *n*-th transaction for cloud providers r, as shown formula (12) below.

$$gT_r^n = \rho * sT_r^n \tag{12}$$

The algorithm of whole trust model above uses pseudocode to describe as following.

Algorithm 1: program GlobalTrust ($w_r^1, w_r^2, w_r^3, TL_{Pr}^{n-1}, RES_{Pr}^{n-1}, SU_{Pr}^{n-1}, sT_{r-1}^n$, Output gT_r^n)

```
begin
```

$t = t_{current} - t_{previous}$;

$\rho = e^{-\beta * t}$;

$w_r := (w_r^1, w_r^2, w_r^3)$

$DF_{Pr}^n := W_r * (V_{Pr}^n - V_{Pr}^c)$

```
if (DF_Pr^n > 0) then
```

$d := 1$

$$sT_r^n = sT_{r-1}^n + \frac{1 - d * sT_{r-1}^n}{\mu} ;$$

```
else
```

$d := \gamma^{DF_{Pr}^n}$;

$$sT_r^n := sT_{r-1}^n + \frac{1 - d * sT_{r-1}^n}{\varphi} ;$$

```
end if
```

$gT_r^n = \rho * sT_r^n$;

```
end.
```

5 Experiments and Analysis

In this section, we carry out the simulations to verify the trust model presented in previous sections. The simulation experiment was carried out on the CloudSim [20-21] and implemented with Java language. CloudSim certainly supporting the modeling and simulation of the infrastructure of a large scale cloud computing is a self-contained platform for supporting datacenters, service agents, management and distribution of strategies. Its characteristics mainly include the following two aspects: virtualization engine, which is designed to help build and manage multiple, independent, collaborative virtual services on a datacenter node; the flexible handover can be provided between the time sharing and space sharing in the process of distribution of processing cores for the virtualization service.

Simulation results verify the validity of the model. There are two entities, cloud consumer entity (CSC) and cloud provider (CSP) entity in the cloud computing. Through simulation of cooperation between the cloud provider and cloud consumer, the changes in the trust value for providers of different strength were explored; the performance of multiple clouds was verified, and finally, time factor was verified on the effect of trust value. Parameter settings were shown in Table 1.

Table 1. Parameter initialization list

parameters	initialization	description
ψ	10	penalty factor
μ	10	reward factor
β	0.005	time decay rate
sT^0	0.5	dynamic trust value
γ	2	fulfillment decay rate

Exp1: to verify the changes of trust value for different cloud providers. In order to verify the change of the trust value of different cloud providers, in this experiment, we conducted 50 transactions respectively for the five cloud providers to observe the changes of trust value. After the 50 transactions, we observed the change of the trust value. Their initial trust value is 0.5 respectively, and the changes of trust value were shown in Fig. 3 after 50 times transaction with cloud consumers.

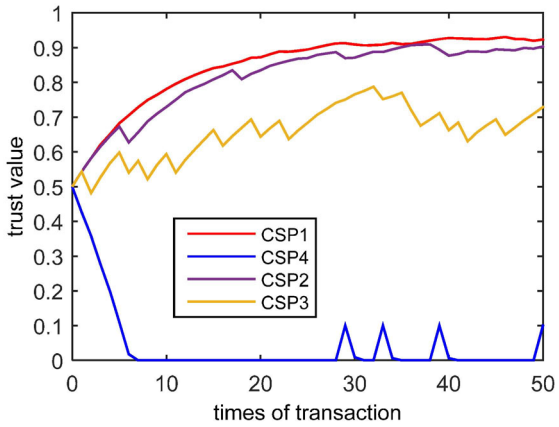

Fig. 3. Performances of CSP

Among them, CSP1 always provides good service, and actively trades with the cloud consumer so its trust value has been on the rise in the overall state after 50 transactions; CSP2 provides good service, but its activity to trade with the cloud consumers is not high, although its trust value has been in the rising state and is lower than the trust value of CSP1; CSP3 provider is at random, sometimes provide a reliable service, and sometimes not very good fulfillment, so its trust value is lower than the CSP1 and CSP2; CSP4 provider provides the bad service far below the SLA service agreement, so its trust value is at a low level after 50 times transactions.

Exp2: to verify the performance of multiple clouds. In order to verify the performance of multiple clouds, in this experiment, we respectively conducted five transactions in case of the single cloud and

multiple clouds (MCTM1 represents that the consumers use the service of CSP1 in case of multiple clouds, so does MCTM2, MCTM3; SCTM1 represents that the consumers use the service of CSP1 in case of single cloud, so does SCTM2, SCTM3). As shown in Fig. 4.

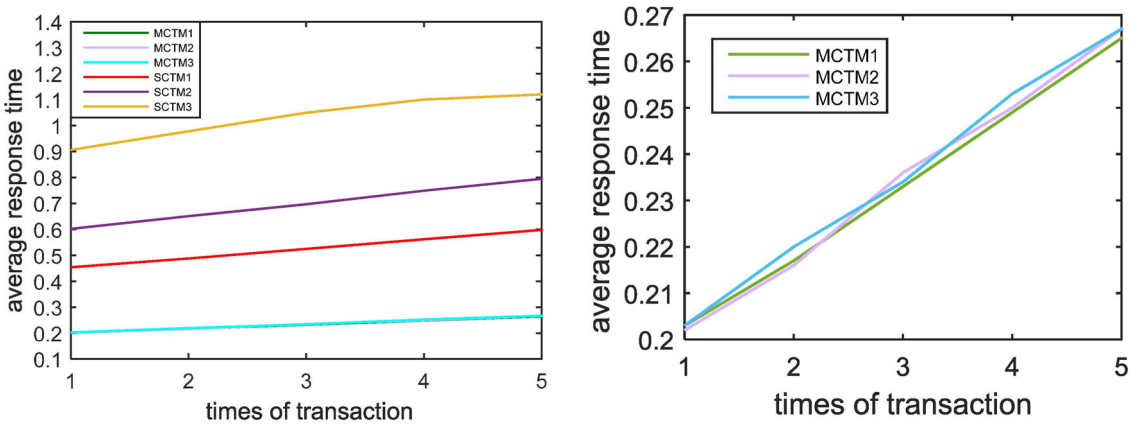

Fig. 4. Performances of multi-clouds

It can be seen from Fig. 4, because of multiple clouds, the number of tasks sent to each provider was less than the number of tasks of using single provider, so the average response time was less than that of single cloud.

Exp3: to verify the time factor. In the process of trust evaluation, the time have certain impact on trust value, and trust value will gradually decrease if a cloud provider has not interacted with cloud consumers for a long time. In the trust model, the update of trust value is necessary, we design the time factor to make the trust value more reasonable. Taking CSP1 provides as example, CSP1 is a provider that provides true and reliable provider and actively transacts. This experiment compared the changes of trust value of CSP1 provider after trading many transactions with cloud consumers in the case of considering the time factor and not considering the time factor. A total of 50 times transactions were carried out to observe the change of trust values. Fig. 5 shows the results.

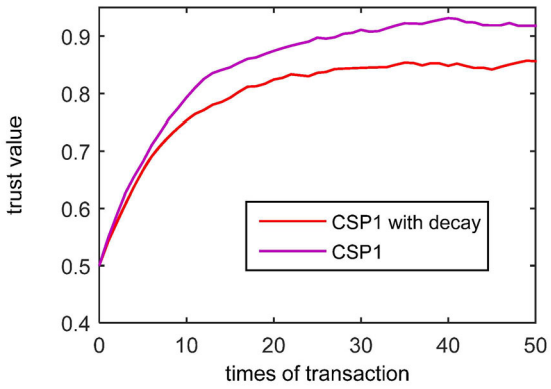

Fig. 5. The impact of time factor

According to the figure, the trust value of CSP1 providers under the condition with time decay and without time decay was compared. As can be seen from the figure, with the increasing of the number of transactions the trust value is always on the rise under the both cases, but the trust value of CSP1 provider with decay grew slower than CSP1 provider. We can conclude that the shorter the interval of transactions is, the higher the trust value is.

6 Conclusions

This paper presents SLA trust model for multiple clouds, the transaction process is monitored to prevent dishonest behaviors of providers (such as providing fake services, failing to fulfill SLA agreement), the

calculation of trust value takes an iterative approach, and the decay model was integrated to the calculation of trust value. However, the model only considers a provider provides services for a cloud consumer while the situation where a provider provides services for multiple cloud consumers does not be considered, which will be considered further.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61402097, the Natural Science Foundation of Liaoning Province of China under Grant No. 201602261, and the Fundamental Research Funds for the Central Universities under Grant No. N151708005.

References

[1] M. Alhamad, T. Dillon, E. Chang, Conceptual SLA framework for cloud computing, in: Proc. Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on, 2014.

[2] M. Alhamad, T. Dillon, E. Chang, Sla-based trust model for cloud computing, in: Proc. Network-Based Information Systems (NBiS), 2010 13th International Conference on. IEEE, 2010.

[3] M. Alhamad, T. Dillon, E. Chang, Service level agreement for distributed services: a review, in: Proc. Dependable, Autonomic and Secure Computing (DASC), 2011 Ninth International Conference on IEEE, 2011.

[4] A.S.A. Saleh, E.M.R. Hamed, M. Hashem, Building trust management model for cloud computing, in: Proc. Informatics and Systems (INFOS), 2014 9th International Conference on. IEEE, 2014.

[5] X. Li, J. Du, Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing, IET Information Security 7(1)(2013) 39-50.

[6] A. Das, M.M. Islam, SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems, IEEE Transactions on Dependable and Secure Computing 9(2)(2012) 261-274.

[7] M. Wang, X. Wu, W. Zhang, F. Ding, J. Zhou, G. Pei, A conceptual platform of SLA in cloud computing, in: Proc. Dependable, Autonomic and Secure Computing (DASC), 2011 Ninth International Conference on IEEE, 2011.

[8] P. Manuel, A trust model of cloud computing based on quality of service, Annals of Operations Research 233(1)(2015) 281.

[9] A. Mohsenzadeh, H. Motameni, A trust model between cloud entities using fuzzy mathematics, Journal of Intelligent & Fuzzy Systems 29(5)(2015) 1795-1803.

[10] T.H. Noor, Q.Z. Sheng, Z. Maamar, S. Zeadally, Managing trust in the cloud: state of the art and research challenges, Computer 49(2)(2016) 34-45.

[11] O. Jules, A. Hafid, M.A. Serhani, Bayesian network, and probabilistic ontology driven trust model for sla management of cloud services, in: Proc. Cloud Networking (CloudNet), 2014 3rd International Conference on IEEE, 2014.

[12] Z.H. Tan, X.W. Wang, W. Cheng, G.R. Chang, Z.L. Zhu, A distributed trust model for peer-to-peer networks based on multi-dimension-history vector, Jisuanji Xuebao (Chinese Journal of Computers) 33(9)(2010) 1725-1735.

[13] Z. Tan, X. Wang, X. Wang, A novel iterative and dynamic trust computing model for large scaled P2P networks. <https://www.hindawi.com/journals/misy/2016/3610157/>, 2016.

[14] Y. Wang, J. Vassileva, Trust and reputation model in peer-to-peer networks, in: Proc. International Conference on Peer-To-Peer Computing. IEEE Computer Society, 2003.

[15] N.K. Saini, V.K. Sihag, R.C. Yadav, A reactive approach for detection of collusion attacks in P2P trust and reputation systems, in: Proc. Advance Computing Conference. IEEE, 2014.

[16] V. Binu, N.D. Gangadhar, A cloud computing service level agreement framework with negotiation and secure monitoring, in: Proc. IEEE International Conference on Cloud Computing in Emerging Markets, 2014.

[17] C. Li, S. Wang, L. Kang, L. Guo, Y. Cao, Trust evaluation model of cloud manufacturing service platform, The International Journal of Advanced Manufacturing Technology 75(1-4)(2014) 489-501.

[18] M. Chiregi, N.J. Navimipour, A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities, Computers in Human Behavior 60(2016) 280-292.

[19] T.H. Noor, Q.Z. Sheng, L. Yao, S. Dustdar, A.H.H. Ngu, CloudArmor: supporting reputation-based trust management for cloud services. IEEE Transactions on Parallel and Distributed Systems 27(2)(2016) 367-380.

[20] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and Experience 41(1)(2011) 23-50.

[21] R.N. Calheiros, R. Ranjan, C.A. De Rose, R. Buyya, Cloudsim: a novel framework for modeling and simulation of cloud computing infrastructures and services. <https://arxiv.org/ftp/arxiv/papers/0903/0903.2525.pdf>, 2009.