International Journal of Applied Science and Engineering
2020. 17, 1: 1-10

Dynamic Multi Attribute Trust Evaluation System for IaaS Services

Alagumani Selvaraj* and Subashini Sundararajan

VIT University, Chennai Campus, India

Abstract: Cloud is a highly distributed environment which enables the sharing of computing resources as services to the users on demand. Trust management plays a vibrant role in distributed computing environments where most of the interactions are likely to be take place in an anonymous manner. The trust management for cloud services is a challenging process due to the dynamic nature and distributed environment of the cloud. As the dynamic nature of cloud paves way for variations in the performance of the service, constant scrutinizing of service attributes is essential for the conformance of service agreement. This paper proposes a dynamic trust evaluation system for cloud services which act as an effective means to make trust judgment using quality of service as attributes. The evaluation process considers time as a major influencing factor in the estimation of trust value and the quality attributes are monitored and quantified at different time slots. The global trust value is derived by the aggregation of trust values at different time slots and that is done dynamically at randomized time intervals. The system adopts dynamic weighting approach to give weightage to the time slots. The system is validated and implemented by setting up a cloud environment and results are discussed.

Keywords: Cloud computing; dynamic trust management; IaaS services; quality attributes; weight calculation.

1. Introduction

Computational and data intensive realms often end up with the scalable storage and infrastructure resource problems. Cloud computing facilitates resource problems by delivering computing, storage and network as services to the computational and data intensive organizations thereby sorting the complications in setting up and maintenance outflows. Cloud computing is circumscribed with five major characteristics such as broad network access, multitenancy, shared resources, scalability and pay-per use model. The most beneficial factor that drives the user to use cloud resources is that the users are given privilege to access whatever they need without worrying about how it was offered. The advantages of cloud computing make small and medium scale businesses to step towards cloud computing solutions. The provisioning of cloud services is made by service provider on requisition from the service user and the service is handed over and taken away dynamically. The users are relieved from the overhead of computing resource maintenance and they are requisite to pay only for the services they have used.

The introduction of cloud services increases the dependency of users on service provider. The mechanism of storing data in a third-party environment and unacquaintance of the data location fosters the question of trust. The resources are shared by many users whose identity is not known.

*Corresponding author; e-mail: alagumani.s2013@vit.ac.in
Received 13 July 2018
doi:10.6703/IJASE.202003_17(1).001

Accepted 13 January 2020

Trust, security and privacy issues in cloud hinder the business to further proceed towards the cloud. The various issues and challenges related to cloud security and trust have been extensively reviewed in different perceptive. In some of the literature, the concept of service broker/ Auditor is introduced to dealt with the trust management process. The role of broker includes the analysis of various services available and helps in service selection process. The auditor is responsible for monitoring the performance of the cloud service. Usually broker/auditor are assigned either by the service provider or by a third party entity who does auditing on behalf of user. The provider assigned auditor/broker will do favor for the service provider. The introduction of third party auditor again leads to the problem of dependency and trust. Moreover, users will be suffered with management overheads of auditing. Therefore, the solution is to integrate trust management as a component in cloud environment.

1.1 Trust Management

Trust management in cloud is a major research area. Even though exhibition of trust is done by extensive tools and systems, still the systems lack a clear picture. Trust mechanisms are broadly classified into four categories: policy based, reputation based, attribute based and SLA based trust mechanism [1]. In policy based trust mechanism, certificates are issued to justify the trust level of the service and the issuance of certificate are done by appropriate certificate authority. Unfortunately, still a formal accreditation mechanism does not exist for the cloud services. The reputation based trust sums up overall opinions about a particular service to estimate the trust value. The collection and aggregation of overall opinions of users is a tedious task as cloud is a heterogeneous environment composed of different services which are used by varied users. The performance of quality of service parameters is considered as evidences in attribute based trust. In SLA based trust, the meeting up of service level agreement between the user and provider is verified.

Trust management in cloud is an inherent and manifold element. The definition of trust varies depending on the context for it has been used. Still, there is no standard trust management system specific to cloud environment. Existing trust models are built on top of the existing systems which are inefficient when adapted to cloud environment. Dynamic and highly distributed environment of cloud complicates the trust evaluation process. The objective of trust management system is to identify an appropriate cloud solution that meets up the users' requirement. Trust management gives its hand to both the service provider and service consumer. In service provider perspective, it aids in improving its capability and in consumer perceptive it assists in cloud service selection.

1.2 Openstack

The proposed system uses openstack as a middleware to implement cloud environment. Openstack is an open source tool which provides infrastructure solution to set up a private cloud [2]. There are six projects to handle core services in cloud: nova (compute), Neutron (networking), swift (storage), keystone (identity) and glance (image). Nova provides support for the creation of instances and is responsible for the management of instances. Neutron provides networking as a service among various openstack interface services. Swift is an openstack storage project which is responsible for storing and retrieving data through application interfaces. Glance provides image services which deals with registration and retrieval of virtual machine image. Keystone provides identity service which is responsible for authentication and authorization of tenants through interface. Figure 1 shows the components and integration of openstack services.

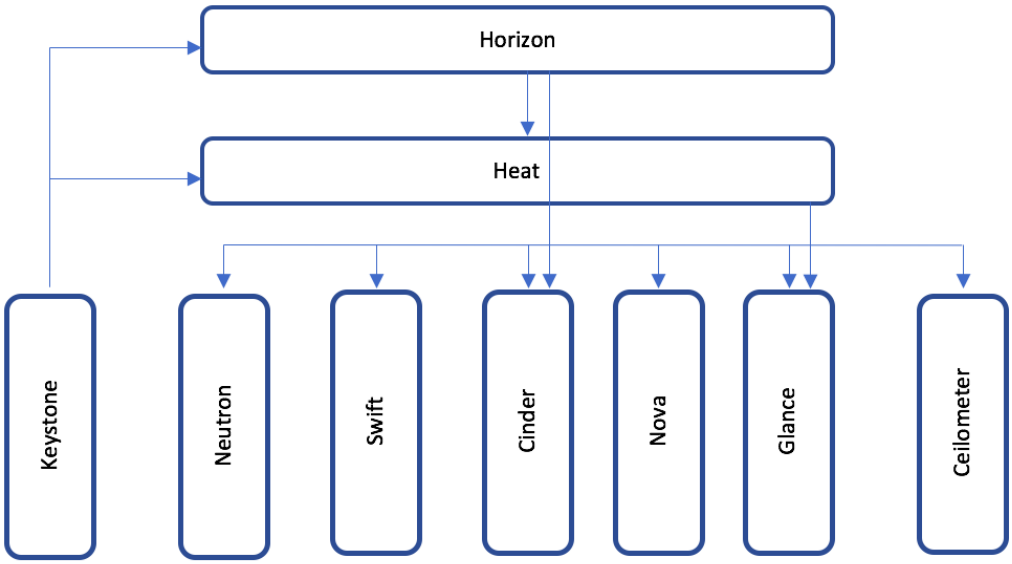

Figure 1. Openstack architecture.

This paper focuses on the trust evaluation method which derives trust value based on quality of service. The paper is outlined in the following six sections. The section 1 presents introduction about cloud and trust management system. In section 1, a brief discussion about need for trust in cloud is done. The state-of- art on existing cloud trust models is carried out in section 2. The proposed work is explained in section 3 and section 4 covers the implementation details of proposed work. The section 5 discusses the results obtained by implementing the proposed system. The conclusion and future work is stated in section 6.

2. Related Work

Trust management in cloud is a major research topic in recent years. The trust issues related to cloud is highlighted in the survey made by Dawei Sun et. al [3]. The survey aims in analysing the security, privacy and trust threats and discusses the ways to eliminate them by providing a highly secure and trustworthy cloud environment. Jingwei Huang et. al. [1] carried out an intensive analysis of the existing trust management mechanisms and came up with a suggestion of policy based approach to carry out the formal auditing and an evidence based approach to believe the service based on the attributes as evidences.

Existing state-of-art for cloud service selection methodologies addresses the ranking of cloud service, hinge on their performance and selection system which deals with user requirements. Initially, the user was provisioned with the service on First In First Out (FIFO) basis without considering the trust value. Paul Manuel and Tamarai selvi [4] proposed combined trust, a novel method which combines three trust model such as identity-Based Trust, capability-based trust and behavior-based trust. The system utilized the cloud broker whose authorization and authentication were implemented using Kerberos. Paul Manuel calculated the trust value by considering four parameters such as availability, reliability, turnaround efficiency and integrity [5]. In both the QoS model trust and combined trust model, simple addictive function is used which uses static weights for the trust value calculation.

Le Sun et. al. [6] made a survey on various multi criteria based decision system for cloud service selection by looking into five factors such as decision-making methods, data representation of cloud selection approach, parameters and characteristics of Cloud services, contexts and purposes. Through the survey, issues related to service selection were identified. The issues were related to lack of standard service registry, lack of standard parameter list satisfying different user requirements, inadequate dynamic service performance monitoring system, problem in dealing with uncertain behaviour of attributes and difficulty in handling relationship between attributes. Shangguang Wang et. Al. [7] pondered monitoring of high variance quality of service data and evaluate the service provider capability based on user preference. To deal with uncertainty of quality attributes, fuzzy logic was employed. Mayank Kumar Goyal et. al. [8] used five QoS parameters such as initiation time, price processing time, fault rate, default and bandwidth to carry out trust evaluation. The parameters are monitored at different time and the trust value calculated were updated at fixed time interval to show the changes in performance. The system failed to incorporate the weightage to the time period as there exhibit the uncertainty in performance over different time period. In addition, the existing systems used fixed interval which is inefficient for cloud environment.

Xiaoyong Li and Junping Du [9] presented an adaptive trust management system which uses the rough set theory and induced ordered weighted averaging operator to verify the conformance of SLA. Though the model provided weightage to the time slot, it lacks the application of preference to the attribute values. Mingdong Tang et. al. [10] proposed an integrated trust evaluation method where objective and subjective factors are combined to perform trust assessment. The method uses QoS monitoring and user feedback as objective and subjective factors in deriving trust score. The evaluation system acted as a middleware which incorporated the identification of malicious user feedback. Shuai Ding et. al. [11] proposed a trust framework which an utility mapping is done between user preference and the quantitative attributes. The systems provided platform to combine the user's feedback and service performance but did not address the dynamic nature. Sarbjeet Singh et. al. [12] developed a compliance based trust evaluation framework which enables the evaluation of trustworthiness of a CSP using the improved TOPSIS method. Chenhao Qu and Rajkumar Buyya [13] proposed a hierarchical fuzzy logic trust management system which combines quality of service with user expectation. The model used fuzzy engine which took user preference as input and output the uncertainty using linguistic descriptions.

From the existing works, it has been observed that the existing trust systems did not efficiently cover up the dynamic nature of service. The monitoring of services and updating of trust value was done at fixed interval providing loop hole for the fraudulent service providers. Therefore, it is important to evaluate the trust based on attributes of services and evaluation of trust value is done dynamically.

3. Proposed System

The system proposes a dynamic trust evaluation system which uses quality of service as attributes in making trust judgment. The system function includes QoS monitoring service, prioritization service and trust evaluation. The architecture of proposed system is depicted in Figure 2.

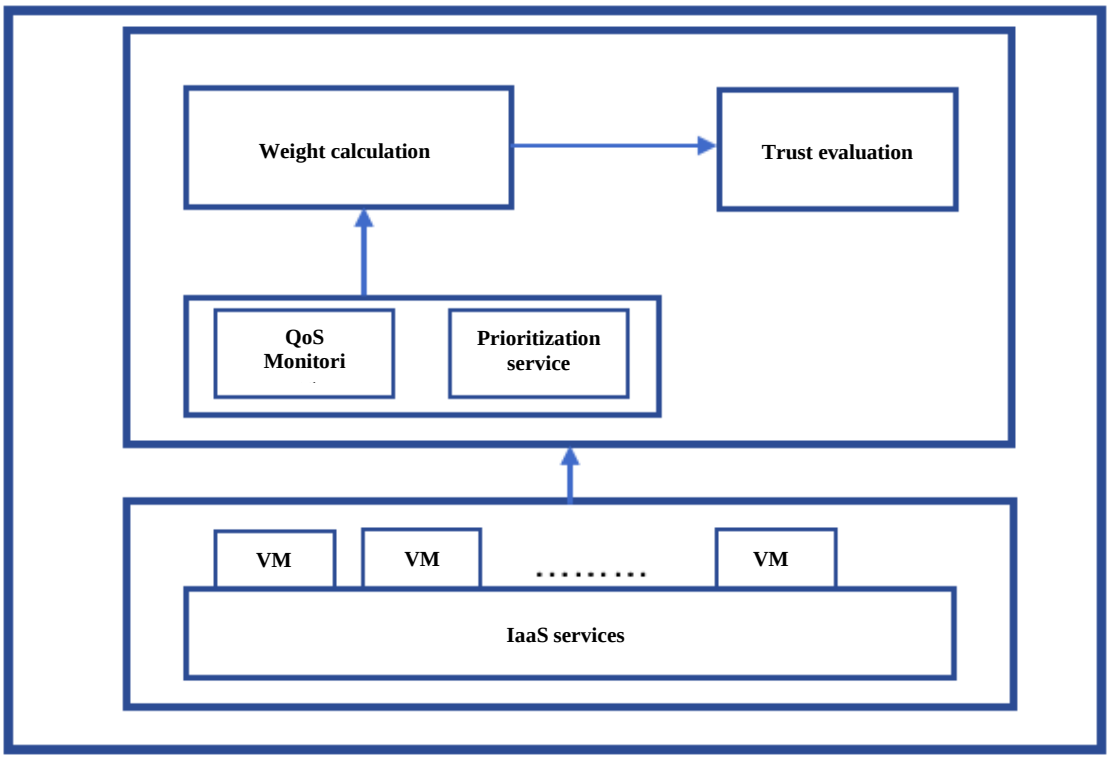

Figure 2. Architecture of Dynamic trust evaluation system.

3.1 QoS Monitoring

The system uses quality attributes to evaluate the trust value. The QoS attributes defined in Service Measurement Index (SMI) framework are used in the proposed system [14]. The new metrics can be easily added and inappropriate attributes can be deleted easily. The attributes are represented in a tree like structure and follow a bottom up approach in deriving the trust value. The metrics for leaf nodes are defined in SMI framework. The values of leaf nodes are summed up to derive the value of upper level nodes. The leaf attributes are classified as quantitative and qualitative attributes. The qualitative attributes such as security and cost are static in nature and they are assigned values based upon the level of standard. The quantitative attributes exhibit dynamic behavior and values are measured according to the defined metric. Figure 3 summarizes the quality attributes and their metrics.

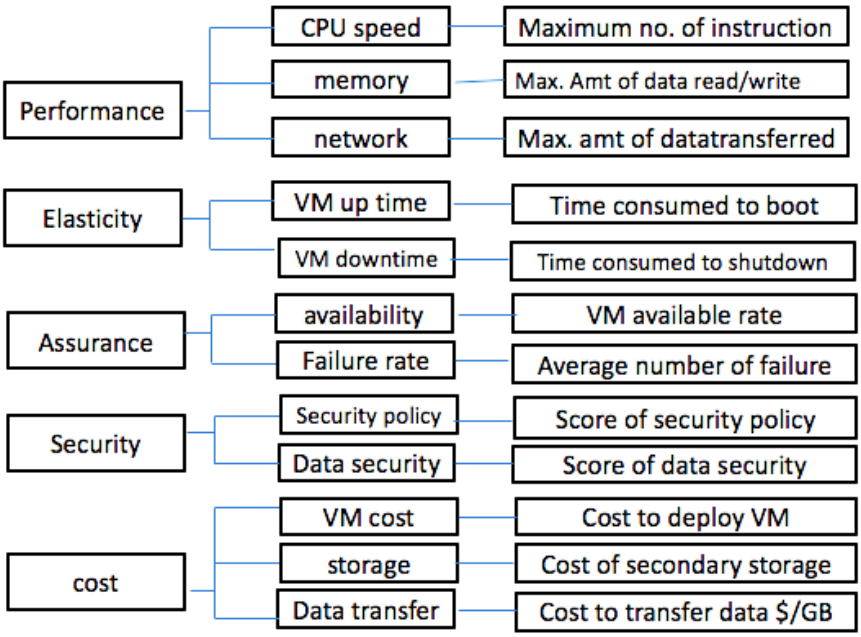

Figure 3. Attributes of IaaS service.

3.2 Prioritization Service

The data values are collected at different time spots. Due to the dynamic nature of cloud environment, the monitoring of attributes is done at various time spots. The selection of time spots can be done in three ways: fixed time interval, random interval, load-based interval. In fixed interval, the attributes values are collected at regular time interval that is for every 5 mins, one hour, 12 hours. Time spots are picked up randomly that is the time interval is not fixed. In load based interval, the monitoring is done during peak hours that is when the load is high.

In this system, random based monitoring is carried out. The trust value for a particular time period can be found out by summing up all the time spot values. As cloud shows variations in performance, the dynamic nature initiates the collection of values at different time spot. To consider the freshness of these values, a decision is made depending on the distance between the current time and measured time. The weights to each time slot are distributed such that the weights progressively increase from old to new timeslots.

3.3 Weight Calculation

To deal with dynamicity, more weightage is given to recent transactions compared to the older one. The weight increases gradually from the starting to ending timeslots. This can be achieved by using a magnifying variable α (alpha) whose value lies between 0 to 1. When the value of α=0.5, all the timeslots have equally distributed value. At α=1, the weight for recent timeslot becomes 1 and at α=0, weight to the old timeslot is 1. The weight increases gradually as the value of α gets increased.

3.4 Trust Evaluation

The attributes values are thus collected and stored in a database at every time spot. The trust for a time slot is determined by aggregating the attribute value with the corresponding weight. The weight w_i for the attribute i is calculated using the following formula.

$$W_i = \sum_{j=0}^{i} x_j / \sum x - \sum_{j=0}^{i-1} x_j / \sum x \quad (1)$$

To perform aggregation operation, additive weighted method is used. The trust value at time slot j is calculated as

$$T_j = \sum w_i * a_i \quad (2)$$

Now, the trust value for different time slot is estimated and the global trust value for a time period has to be calculated. The more weightage is given to the recent time slots and the older time slots are given less weightage. The global trust value is calculated as follows.

$$T_g = \sum w_j * T_j \quad (3)$$

Algorithm 1. Trust evaluation.

Initialization

Let $S = \{s_1, s_2, \ldots, s_n\}$ denotes a set of services

$A = \{a_1, a_2, \ldots a_n\}$ be the set of attributes

$W = \{w_1, w_2, \ldots w_n\}$ be the set of weights

$t = \{t_1, t_2, \ldots, t_n\}$ be the random time slot

Attribute monitoring

for i= 1 to n

$E_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$

Weight Calculation

w_1 : $w_1[(n-1)\alpha+1-n*w_1]^n = [(n-1)\alpha]^{n-1}[(n-1)\alpha-n)w_1+1]$

w_n : $\frac{((n-1)\propto-n)w_1+1}{(n-1)\propto+1-n*w_1}$

for j = 2 to n-1

w_j : $\sqrt[n-1]{w_1{}^{n-j} w_n{}^{j-1}}$

Dynamic trust calculation

Trust value for a time slot T is given by

for j=1 to n

$T_t = \sum T_i * w_j$ where i=1,2,….,m and j= 1,2,…,n

4. Implementation

The proposed system is implemented in openstack which is used to deploy infrastructure as a service for cloud environment. The openstack comprises of nine organized components which operate to govern compute, storage and network resources. The system uses three node architecture (controller node, network node and compute node) with optional storage nodes. The test environment was built on virtual machines. The benefits of using virtual machines include support for multiple nodes by single server machine and ability to roll back using snap shot option. The identity service and image services are implemented in controller node which is also responsible for managing compute and network node. The network node manages the networking services such as routing, switching, NAT and DHCP. The instances are maintained by compute node which uses KVM as hypervisor.

4.1 Data Collection

In addition to the instance management, compute node runs telemetry service which is responsible for the collection of instance metrics. The agents in telemetry components are central agent, a collector and a API server. The central agent polls for the statistical data of resource utilization at various time intervals. The collector agent collects the polled data and stored it in a data base. The API server is responsible for accessing the data stored in the data base through the dashboard services. New metrics can be added and deleted by configuring the telemetry service. The data thus collected are given as input for the trust evaluation system for the derivation of trust value.

4.2 Trust Calculation

The attributes values were collected at different time slots. The attributes are measured using the metrics given in the SMI framework. On collecting the metric, the values are stored in a database for different time slot. The normalization of data values is done to have a standard value in the range of 0 to 1. The two weight vectors are calculated, one for the trust value evaluation at every time slot and the other to give prioritization to recent time slot. The trust value is calculated by aggregating the attribute values with the corresponding weights. For aggregation, weights are used which are derived from the attributes values. The trust value is calculated for each time slot. The value thus collected are summed up with associated weight to derive the trust value for the time period.

5. Results and Discussion

The proposed trust evaluation model is implemented and tested by setting up a private cloud environment using openstack. The virtual machine with different attributes specification are created and the virtual machines attributes are monitored at different time slot. The collected data are stored in a database. Table 1 represent the attribute values for the different service at the time spot t. The normalization of data is performed to make a common scale for all the attributes. The trust value is calculated on each time spot.

Table 1. Attribute value at a time spot.

Service/att	attr1	attr2	attr3	attr4
S1	7.95	784.28	26.09	29.65
S2	6.46	1830	46.05	53.25
S3	0.065	493.64	76.4	57.03
S4	0.061	405.39	59.04	33.95
S5	0.065	469.62	76.35	58.18
S6	0.016	183.81	0.353	161.43
S7	0.061	219.52	0.282	0.132

The weight vector is calculated by implementing the proposed algorithm and the sample weights are tabulated in Table 2.

Table 2. Weight vector.

	a = 0.5	a = 0.6	a = 0.7	a = 0.8	a = 0.9
w_1	0.2500	0.4167	0.4938	0.5965	0.7641
w_2	0.2500	0.2334	0.2373	0.2520	0.1821
w_3	0.2500	0.1309	0.1138	0.1065	0.0435
w_4	0.2500	0.0735	0.0549	0.0450	0.0104

Table 3 tabulates the trust value for the service at particular time spot. The global trust value for a time slot is arrived by aggregating the trust value at various time spot with associated weights.

Table 3. Trust value for services.

service/att	attr1	attr2	attr3	attr4	TRUST
S1	1	0.3647	0.3390	0.1830	0.4717
S2	0.8122	1	0.6012	0.3293	0.6856
S3	0.0061	0.1882	1	0.3527	0.3867
S4	0.0056	0.1346	0.7719	0.2096	0.2804
S5	0.00617	0.1736	0.9993	0.3598	0.3847
S6	0	0	0.00093	1	0.2502
S7	0.00567	0.02169	0	0	0.00684

6. Conclusion

In this paper, a dynamic multi attribute trust evaluation system is proposed which considers quality of service as attributes to obtain the trust value. The attribute values are collected at random timeslots to deal with the dynamic behaviour of cloud services. The aggregation of different time slot values is done to come up with the ideal service. The weightage is applied for different time slots and dynamicity is applied in the derivation of weight. The system is implemented using openstack cloud environment and the results were discussed. The proposed system aids the user in finding an optimal service based on quality of service. Furthermore, the weights allotted to the timeslots plays a foremost role in service selection. The proposed system evaluated the trust value based on past interactions and the future work will extend the system to perform prediction analysis by analysing the past performance.

References

[1] Huang, J. W. and Nicol, D. N. 2013. Trust mechanisms for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 2-9.

[2] https://docs.openstack.org/mitaka/install-guide-ubuntu/: accessed on August, 2017.

[3] Sun, D. W., Chang, G., Sun, L. and Wang, X. W. 2011. Surveying and Analyzing Security, Privacy and Trust Issues in Cloud Computing Environments. *Science Direct, Procedia Engineering*, 15:2852-2856.

[4] Manuel, P. D., Selvi, S. T. and Barr, M. I. A .E. 2011. A Novel Trust Management System for Cloud Computing - IaaS Providers. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 79:3-22.

[5] Manuel, P. 2015. A trust model of cloud computing based on Quality of Service. *Annals Of Operations Research* , 233, 1:281-292.

[6] Sun, L., Dong, H., Hussain, F. K., Hussain, O. K. and Chang, E. 2014. Cloud service selection: State-of-the-art and future research directions. *Journal of Network and Computer Applications*, 45:134-150.

[7] Wang, S. G., Liu, Z. P., Sun, Q., Zou, H. and Yang, F. C. 2014. Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. *J Intell Manuf*, 25:283-291.

[8] Goyal, M. K., Gupta, P., Aggarwal, A. and Kumar, P. 2012. QoS Based Trust Management Model for Cloud IaaS. *2nd IEEE International Conference on Parallel, Distributed and Grid Computing.*

[9] Li, X. Y. and Du, J. P. 2013. Adaptive and attribute-based trust model for service- level agreement guarantee in cloud computing. *IET Inf. Secur*, 7, 1:39-50.

[10] Tang, M. D., Dai, X. L., Liu, J. X. and Chen, J. J. 2017. Towards a trust evaluation middleware for cloud service selection. *Future Generation Computer Systems,* 74:302-312.

[11] Singh, S. and Sidhu, J. 2017. Compliance-based Multi-dimensional Trust Evaluation System for determining trustworthiness of Cloud Service Providers. *Future Generation Computer Systems*, 67:109-132.

[12] Ding, S., Yang, S., Zhang, Y., Liang, C. and Xia, C. 2014. Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. *Knowl.-Based Syst*, 56:216-225.

[13] Qu, C. and Buyya, R. 2014. A Cloud Trust Evaluation System using Hierarchical Fuzzy Inference System for Service Selection. *IEEE 28th International Conference on Advanced Information Networking and Applications*.

[14] Garg, S. K., Versteeg, S. and Buyya, R. 2013. A framework for ranking of cloud computing services. *Future Generation Computer Systems*, 29, 4:1012-1023.