International Journal of Engineering

Journal Homepage: www.ije.ir

Bandwidth and Delay Optimization by Integration of Software Trust Estimator with Multi-user Cloud Resource Competence

S. M. Mirrezaei*

Faulty of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran

P A P E R I N F O

Paper history:
Received 27 March 2020
Received in revised form 02 June 2020
Accepted 08 June 2020

Keywords:
Bandwidth
Behavior Trust Estimator
Cloud Resources
Competence Optimization
Software-as-a-Service
User Feedback

A B S T R A C T

Trust establishment is one of the significant resources to enhance the scalability and reliability of resources on cloud environments. To establish a novel trust model on SaaS (Software as a Service) cloud resources and to optimize the resource utilization of multiple user requests, an integrated software trust estimator with multi-user resource competence (IST-MRC) optimization mechanism is proposed in this paper. IST-MRC optimization mechanism combines its trustworthy properties with optimal resource allocation, on the requisition of the software apps, without any traffic occurrence in the cloud environment. Initially, a behavior trust estimator is developed in the IST-MRC mechanism to measure the trust value of the software service zone. The trust value is estimated, based on Software Availability Rate, Hit Rate, and User Feedback regarding the specific software apps. Next, the resources are optimized to multiple users using competence optimization. The competence optimization in the IST-MRC mechanism computes the processor speed, bandwidth, and latency to handle the varied traffic conditions on multiple user requests. Experiments are conducted to measure and evaluate factors, such as Successful Request Handles, Resource Utilization Efficiency, Latency Time, and Trust Success Ratio on the multiple users.

***doi**: 10.5829/ije.2020.33.07a.04*

1. INTRODUCTION

The ever-increasing and broadening limelight, tendered to the cloud computing environment, presents a natural means to expand the potential of content delivery networks to sustain the progress of the content linked to the text, image, and the like. The dynamic request redirection (DRR-CCMN) of cloud-centric media network [1] considers the drawback of redirecting the users' requests to multiple destination virtual machines (VMs) optimally and most favorably to lessen the cost of computation. The request arrival process is also geared up in such a way that it can effectively switch on between the two models - the normal and flash crowd. Nevertheless, DRR-CCMN has been substantiated to be unproductive when it is used between multiple dispatchers, with diverse traffic circumstances.

With umpteen intricacies mixed up with the distinct traffic conditions, one of the essential requirements for the accomplishment of the clouds, lies in the efficient organization of the task of the cloud virtual machines. The existing cloud schedulers not only overlook the overall cloud infrastructure but also ignore the entire infrastructural properties and, in doing so, end up with various security-related issues, hitches in the privacy of the data, and so on. A cloud scheduler, using the OpenStack (CS-OpenStack) prototype [2], presents a novel cloud scheduler, which takes into consideration the requirements of the users as well as the infrastructural properties.

In reference [3], the task of counting the third party auditor (TPA) is initiated to grant genuineness to cloud data storage. The TPA, in turn, verifies the data integrity of the dynamic cloud data storage that resolutely avoids any addition of a client to achieve the economies of scale for the cloud computing environment.

This ensures more protection during the processing of the multiple auditing tasks; however, simultaneously, there are apprehensions of losing control over their data. On account of Internet usage has reached a new high,

*Corresponding Author's Email: *sm.mirrezaei@shahroodut.ac.ir* (S. M. Mirrezaei)

Please cite this article as: S. M. Mirrezaei, Bandwidth and Delay Optimization by Integration of Software Trust Estimator with Multi-user Cloud Resource Competence, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 33, No. 7, (July 2020) 1192-1200

cloud computing environment affords dynamic scalability. Despite enjoying the privileges given by this new epoch, there are apprehensions of losing grip over the data. So a decentralized information accountability framework [4] is introduced to keep track of the users' data usage in the cloud computing environment via an object-centered approach. Not withstanding the inclusion of proficient and resourceful auditing mechanisms, the component of trust remains unattended.

In trustworthy clouds (TClouds) [5], the focus is on constructing trust models that include different levels of transparency and the establishment of trust. It has been ascertained to be beneficial and advantageous to both the cloud providers and the users; yet, here too, the procedure of evaluating or measuring the trust values continues to be left unaddressed.

In literature [6], a cloud-based information repository is used, which significantly concentrates on the notable aspects of the cloud computing environment, which amalgamate the users' health data and make an accessible analysis of the data, using a trusted third party. As a result, a trust framework [7] is evolved to obtain a broad set of parameters to, not only ascertain the trust, but also includes the cost factor that enables it to establish and negotiate. It includes a trust model to prop up the policy integration, using a high-powered approach.

Internet appliances compass electronic commerce, interactions between the users through electronic mail, and so on; nevertheless, there is still a higher level of unease regarding the trustworthiness of these types of devices. To boost the level of security, in literature [8], a trust evaluation model has been devised based on D-S evidence theory with sliding windows, using the cloud computing environment. It is endowed with opportunities to effectively identify malicious users and offer reliable information in making correct decisions regarding the system. However, the collision behavior remains unanswered.

In general, the framework applies to most of the trust-based models, shared on the software apps, where multiple users evaluate the software trust model utilizing the trust success ratio. Based on techniques as mentioned above and methods, I propose an integrated software trust estimator with a multi-user resource competence (IST-MRC) optimization mechanism to provide a novel trust model in software as a service (SaaS). Trust is the estimation of the cloud software environmental values, and the software trust model of IST-MRC mechanism evaluates the trust value of the resources, based on the observed behavior of the cloud computing environment. With this, the trust management mechanism in the IST-MRC reduces the complexity level of cloud service provisioning. As a result, the resource utilization in the IST-MRC mechanism provides entirely distributed resources to its multiple users, without any traffic occurrence.

IST-MRC optimization mechanism combines its trustworthy properties with optimal resource allocation, on the requisition of the software apps, without any traffic occurrence in the cloud environment. This framework, besides using the behavior-based trust model, utilizes competence optimization to enhance its resource utilization efficiency. Extensive experiments exhibit that the optimization mechanism is equipped with a superior level of successful requests on the users with less Latency Time.

The main contributions of this paper are the introduction of:

- The proposed model uses an integrated software trust estimator with a multi-user resource competence (IST-MRC) optimization mechanism and leads to overcome certain drawbacks of the existing trust models in the cloud computing environment.
- IST-MRC mechanism uses the competence optimization method to utilize the higher processing speed of RAM with higher bandwidth for several specific software apps.
- The proposed optimization mechanism estimates the degree of resource utilization efficiency, based on the plea placed by the cloud user. Therefore, it evaluates the trust success ratio in a resourceful manner.
- To maximize the resource utilization efficiency on the multiple users, the processor speed, bandwidth, and latency to handle the wide-ranging traffic conditions on the multiple user requests are deliberated. According to the results obtained my approach is apt for multiple users with its wide-ranging traffic conditions.

The remaining sections are systematized as follows. In Section 2, an overview of the related works on the trust-based model in a cloud computing environment is deliberated. Section 3 analyzes the general outline of problem descriptions and lays on solutions accordingly. Section 4 elucidates my proposed trust-based model employing multiple users. Section 5 interprets the experimental results pertained to the validation of the performance of the proposed method. Section 6 puts across the concluding observations.

2. RELATED WORKS

Despite attaining tremendous insights, the primary concern being security, cloud computing is still at its early stage. In literature [9], security, trust, as well as the privacy of a cloud computing environment, has been addressed. An elaborate analysis has been executed on these three imperative factors; however, their deployment in the real-time environment continues to be a significant concern, and the time factor has not been scrutinized in a broader sense as well. In reference [10], a novel framework has been designed to support the application of knowledge discovery in a cloud computing

environment, and it ably predicts its application execution time, using Rough Sets Theory; nevertheless, to a greater extent, the uncertainty still exists. Though the cloud computing environment is more suitable for representing trust, in the presence of dynamic updates, it is unwise to include trust. Strategic provisioning of cloud computing services [11] has been initiated to prefer amongst the paramount services. Lack of standard specification and low ratings are some of the issues, which persist. In reference [12], a novel, unfair rating filtering method is set up to furnish reputation and accordingly to opt for the most excellent services.

With the mounting utilization of the Internet, and its applications as well as the resources being delivered to the cloud users, on-demand, data security and privacy concerns have to be given prime attention to increase the users' responsiveness. In literature [13], a comprehensive comparative analysis was made to address the issues related to data security and privacy protections in the cloud computing environment. To curtail the consumption of energy in the cloud data center, virtual machine allocation algorithm [14] is put forward, using efficient resource allocation and particle swarm optimization (PSO) method. Even though the complexities mixed up with the computation are addressed, only, certain resources like CPU and disk have been incorporated. To address the challenges involved in the security aspect of the cloud computing environment, cloud authentication [15] is to be enhanced to provide a resourceful mechanism against a phishing attack in the initial stages.

The cloud computing environment is a framework that enables on-demand network access amongst its various users. One of the major features to be well-thought-out related to the quality of service is reliability, which has to be dealt with competently. An efficient trust assessment [16] is introduced before the sharing of the infrastructure, and it results in scalable and reliable service for its legitimate users.

One of the highest concerns related to cloud data storage is its inefficient data verification procedure and its reliability; it is a holdup in endorsing trust in the cloud computing environment together with SaaS [17-20].

In reference [21], cloud broker and its need in the related cloud environment are discussed. Also, cloud brokering frameworks of interconnected cloud environments are reviewed. Its authors presented a taxonomy of cloud brokering techniques and comparative analysis of cloud brokering techniques.

Xie et al. [22] proposed a trust model for a container cloud environment, which uses direct trust, recommendation trust, and cooperative trust to calculate the comprehensive trust degree in three trust ways. The results of the simulation experiments showed that the model can effectively solve the trusted problem in the container-based cloud.

Another article in this field is an attempt to address the issue of privacy of big data distributed in the field of cloud computing [23]. One of the most essential data privacy issues is authentication and protection of proprietary data. In the article above, this privacy issue is analyzed by Petri net modeling.

Nowadays, due to the advances in mobile and wireless communication, mobile devices are widely used in daily life. Meantime, in mobile devices, there exist diverse applications that are developed to satisfy the various requirements of mobile users. To relieve this problem, driven by edge computing, the central units (CUs) in fifth-generation wireless systems (5G) could be enhanced into edge nodes (ENs) for processing [24].

Cloud computing, the long-standing dream of computing as a tool, can transform a large part of the information technology industry and make the software even more attractive as a service and shape the way IT hardware is designed and purchased. The authors of [25] provided an architectural framework and principles for energy-efficient cloud computing environments. They call the resource allocation algorithms the load-power-aware in this architecture. The algorithm uses a heuristic to dynamically improve energy efficiency in the data center while guaranteeing the QoS.

3. OVERVIEW

3. 1. Problem Statement The proposed in this paper uses the IST-MRC optimization mechanism, and it is set up to overcome certain drawbacks of the existing trust models in the cloud computing environment. Also, it is apt for multiple users with its wide-ranging traffic conditions. Before going into the facets of the proposed software trust model, it is requisite to define the scope of the study.

Cloud infrastructure is an archetype that primarily concentrates on data sharing, information, or resources over a massive network. The foremost notion behind cloud infrastructure is to emphasize computing with the storage available at any time and at anywhere. With the increasing use of internet-based cloud infrastructure, the availability of the two most vital elements - trust and security - becomes indispensable. An appropriate trust model provides optimal utilization of the resources, under different traffic conditions.

3. 2. Overview of the Entire Software Trust Model In this section, the software trust model is briefly addresed. The software service provisioning IST-MRC is shown in Figure 1. In general, the framework applies to most of the trust-based models, shared on the software apps, where multiple users evaluate the software trust model utilizing the trust success ratio. Trust is the estimation of the cloud software environmental values,

and the software trust model of IST-MRC mechanism evaluates the trust value of the resources, based on the observed behavior of the cloud computing environment. The evaluation of the cloud trust, ultimately, checks whether proper legal materials of the software are provided to the cloud server. The availability of the legal documents, therefore, ensures and establishes mutual trust between the resource providers and the users. With this, the trust management mechanism in the IST-MRC reduces the complexity level of cloud service provisioning. As a result, the resource utilization in the IST-MRC mechanism provides entirely distributed resources to its multiple users, without any traffic occurrence. The software service provisioning, through behavior trust estimator, is represented in Figure 1. Subsequently, the cloud end-users place the request of the needed software apps. The provisioning of the software services, initially, checks the trust value. The trust value is checked, using the IST-MRC mechanism by computing the Software Availability Rate, Hit Rate, and User feedback. The Software Availability Rate, initially, checks whether the licensed software (i.e. the legal documents) is accessible in the cloud zone. Followed by this, the Hit Rate measures the success ratio of the usability between the end-users. To end with, user feedback is also analyzed, using the IST-MRC mechanism to identify the quality of the software service, provided during the transaction.

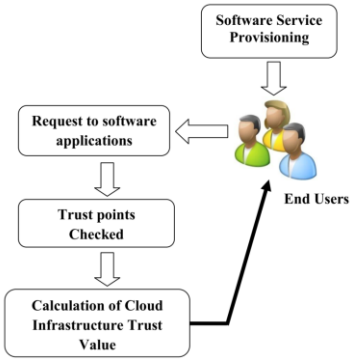

Figure 1. Software service provisioning in the IST-MRC mechanism

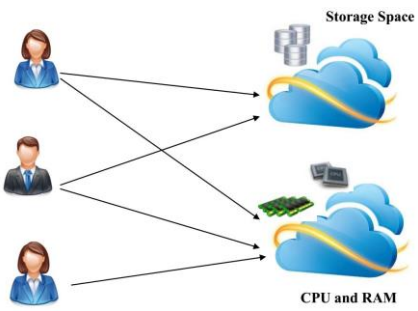

Figure 2. Multiple user requests to cloud resources

The multiple users request the resources for efficient utilization of the software apps for their systems. IST-MRC mechanism uses the competence optimization method to utilize the higher processing speed of RAM with higher bandwidth for several specific software apps. The multiple user requests on cloud infrastructure are depicted in Figure 2. As illustrated in this figure, the resources are made available for the various users, (i.e., User 1, User 2, User 3), depending on the requisition made by the cloud users of the cloud infrastructure. User 1 and User 2 in Figure 2 request for CPU processor and storage space, whereas User 3 requires only a CPU processor for processing the software apps. The whole requests are fulfilled not only with minimal latency time but also without any traffic occurrences.

4. SOFTWARE TRUST MODEL INTEGRATED WITH MULTI-USER RESOURCE COMPETENCE OPTIMIZATION

In this section, the optimization mechanism is described to ascertain a novel trust model and to optimize the resource utilization of the multiple user requests. The proposed optimization mechanism estimates the degree of resource utilization efficiency, based on the plea placed by the cloud user. Therefore, it evaluates the trust success ratio in a resourceful manner. The overall framework of the IST-MRC optimization mechanism is depicted in Figure 3. As illustrated in this figure, the multiple users entreat the cloud server for the software application. The cloud server builds up a trust model in the IST-MRC mechanism using a behavior trust estimator. The behavior of the software apps is perceived, and the trust values are assessed to make out its service provisioning efficiency. In conjunction with this, the trust value is evaluated with a higher scalability ratio, and then the integration work is carried out using the IST-MRC

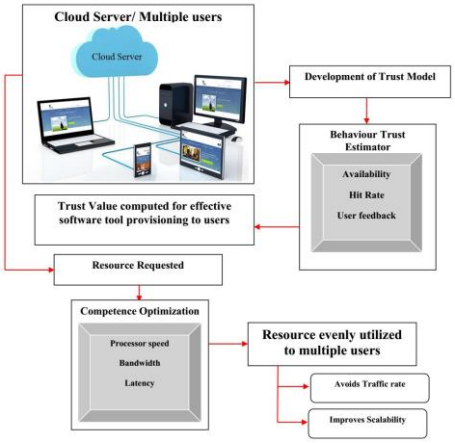

Figure 3. Overall structural diagram of the IST-MRC optimization mechanism

mechanism. The users (i.e. multi-users) in the cloud zone are not provisioned with the valuable resources in an attempt to run the software apps. Consequently, the multiple users implore the cloud zone for the resources to run the application with a higher reliability rate. The IST-MRC mechanism uses the competence optimization method to handle the entire users appeal under different traffic conditions.

4. 1. Behavior Trust Estimator The user behavior trust estimator in the IST-MRC mechanism measures the performance factor of the Availability Rate (A_i), Hit Rate (H_i), and User Feedback Rate (UF_i). The Availability Rate of the software apps in the cloud zone refers to the ratio of the number of times the specific software apps are requested to the total number of times the software has been queried for the operation (i.e., transaction). The availability rate of the application in the IST-MRC mechanism is computed as,

$$A_i = \frac{Number\ of\ times\ the\ software\ requested}{Total\ number\ of\ times\ the\ software\ required} \quad (1)$$

Assume '5' to be the count on the software requested in the cloud zone and '10' to be the number of times the software queried for the transaction.

$A_i = \frac{5}{10} = 0.5$

The availability rate or availability percentage for the specific software apps is subsequently evaluated to be '0.5'. The software hit ratio in the IST-MRC mechanism refers to the ratio of the number of successful operations in the cloud to the total number of operations placed over the specific software.

$$H_i = \frac{Number\ of\ successful\ operations}{Total\ number\ of\ requests\ placed\ over\ the\ operation} \quad (2)$$

To evaluate the hit rate, let us assume that '5' successful operations are carried out in adherence to the '5' successful requests placed over the cloud zone.

$H_i = \frac{5}{5} = 1$

The IST-MRC mechanism achieves a higher hit ratio, without any delay when providing the software services to the end-users. In line with this, the higher hit ratio trust value offers better user feedback to the cloud system. The user feedback UF_i in the IST-MRC mechanism collects the feedback and identifies the trust value, based on the feedback obtained from the repository database. The user feedback ensures the scalability ratio of the cloud zone.

The feedback is dispersed over a range of 0 to 1 in the IST-MRC mechanism, where 1 represents the most trustworthy software collection, and 0 represents the non-trustworthy software collection. The feedback value for the above-computed hit ratio, UF_i is '1', and the feedback varies vigorously, based on the hit ratio attained during the operation. The algorithmic description of Behavior Trust Estimator is described as,

Behavior Trust Estimator
Input: Assume users {$U_1, U_2, U_3 ..., U_n$} for the software apps service requisition
Output: Trust Value computed with Higher Scalability Value
Begin
Step 1: Initially measure the Availability Rate A_i [shown in (1)]
Step 2: Measure the Software Hit Rate H_i [shown in (2)]
Step 3: Final step checks the User Feedback, using (2) Measure Rate
Fuzzy points '0' or '1' is computed in UF_i
Step 4: Sum up the Availability Rate, Hit Rate, and User Feedback values
Compute $BTE = A_i + H_i + UF_i$
End

The algorithm clearly describes the evaluation of behavior trust estimator value through step by step process. The behavior trust estimator is measured as the overall sum of the Availability Rate, Hit Rate, and User Feedback Rate. The behavior trust is formularized as,

$$Behavior\ Trust\ Estimator\ (BTE) = A_i + H_i + UF_i \quad (3)$$

BTE value through the above-illustrated points is expressed as,

BTE = 0.5+1+1= 2.5

The BTE value attained from the assumed points is about '2.5'; finally, the behavior trust estimator achieves a higher scalability rate, using the IST-MRC mechanism, when compared with the existing systems.

4. 2. Competence Optimization The IST-MRC mechanism allocates the resources to the multi-user requests, under different traffic conditions. The resources, such as CPU processor of diverse types, RAM, and storage unit, are allocated to the multiple users, depending on their needs for the efficient processing of the software applications. To avoid any collision during the multi-user requests, the cloud provides the resources with different traffic conditions, using the competence optimization procedure.

The competence optimization computes three important measures, namely the bandwidth, processor speed, and latency, to avoid traffic occurrences throughout requests from multiple users. The main objective of competence optimization in the IST-MRC mechanism is not only to measure the availability of the resources but also to evaluate the required Request Ratio. In given that, it allocates the resources to all of the users without any delay. The competence optimization (CRO) of the resources, using the IST-MRC mechanism is computed as:

$$CRO = \{U_1(P, RAM, Storage\ space), U_2(P, RAM, Storage\ space), U_3(P, RAM)\} \quad (4)$$

From Equation (4), the cloud allocates the resources following the request made by the Users 1, 2, and 3 for the efficient processing of the software application services. The *'P'* in Equation (4) denotes the processor

requests; 'RAM' indicates RAM space needed for the software installation, and *'Storage Space'* is intended for storing the processed results. By reducing the complexity level, every one of the requests is undoubtedly fulfilled without any collision in the IST-MRC mechanism. Besides, the traffic under diverse conditions is handled by computing the speed of the processor and the bandwidth rate, on which the cloud server places the request. By avoiding collisions, efficient Traffic Handling (TH) is obtained using (5).

$$TH = [2*(Pspeed+RAMspeed)]+B/L \quad (5)$$

The optimized resources achieve efficient traffic handling and it is allocated to the multiple users, based on the processor speed, *'Pspeed'* and RAM speed, *'RAMspeed'*. Further, the IST-MRC mechanism sums up the ratio of Bandwidth Rate, *'B'* to the Delay Time, *'L'* to easily tackle all the requests under different traffic conditions. The competence optimization uses the cloud resources to run the software applications in support of the multiple user systems, and it, therefore, attains a higher reliability rate.

5. RESULT AND DISCUSSION

5. 1. Experimental Setup and Analysis The proposed mechanism, the IST-MRC, has been implemented in JAVA with Cloudsim Simulator, and certain statistical results have been obtained to validate it. A particular toolkit is chosen as a simulation platform for easy evaluation of the experimental parameters. Cloudsim holds the cloud structure with its multiple software tools. The machine is simulated with the data center, which comprises a variety of processing speeds, CPU, and RAM. Virtual machines are used for the experimental work in Cloudsim. Cloud computing has emerged as the key technology to deliver consistent, secure, and scalable computational services. The parameters used to obtain the results for simulations are listed in Table 1. TheIST-MRC mechanism is compared against the existing dynamic request redirection (DRRCCMN) [1] in the cloud-centric media network and cloud scheduler using OpenStack (CSOpenStack) [2] prototype. The experiments are conducted with 100 users and 300 requests, as well as with an average BTE of 30. The experiments are used to measure and evaluate factors, such as Successful Request Handles, Resource Utilization Efficiency, Latency Time, and Trust Success Ratio on the multiple users.

5. 2. Performance Evaluation

a. Scenario 1: Successful Request Handles on Multiple Users

To better perceive the effectiveness of the proposed IST-MRC mechanism, substantial experimental results are tabulated in Table 2. The IST-MRC mechanisms compared against the existing dynamic request redirection (DRR-CCMN) [1] in a cloud centric media network and cloud scheduler using OpenStack (CS-OpenStack) [2] prototype. For experimental purposes, Java with Cloudsim simulator is used to experiment with the factors and analyze the measures with the help of the graph values.

Results are accessible for the mixed number of requests, being placed. Request handles on the multiple users measure the ratio of the number of times the software is requested to the total number of requests being made as in (1). The higher, the requests being handled, the more successful the method is. The results disclosed confirm that the successful request handles on the multiple users increase in conjunction with the increase in the number of requests being placed.

TABLE 1. Parameters Setting of CloudSim

Entity Type	Parameters	Value
Tasks (cloudlet)	Length of Tasks	1000-20000
	Total number of Tasks	100-1000
	Priority of the Tasks	High, Medium and Low
Virtual Machine	Total number of VMs	50
	MIPS	500-2000
	VM Memory (RAM)	256-2048
	Bandwidth	500-1000
	Cloudlet Scheduler	Space shared and Time shared
Data Center	Number of PEs requirement	1-4
	Number of Data Center	10
	Number of Host	2-6
	VM Scheduler	Space shared and Time shared

TABLE 2. Tabulation for Request Handles on Multiple Users

No. of Requests Placed (n)	Request Handles on Multiple Users		
	IST-MRC Mechanism	DRR-CCMN	CSOpenStack
20	14	9	7
40	32	18	15
60	52	40	32
80	48	36	30
100	88	72	60
120	105	90	80
140	125	100	85

For experimental purposes, the process is repeated until the 120 requests are fulfilled. Figure 4 illustrates the application handles on multiple users, based on the number of the demands being placed. The IST-MRC mechanism performs relatively well when compared to the two additional methods, DRR-CCMN [1] and CS-OpenStack [2]. The proposed approach has superior modifications, using a trust-based estimator, which eventually examines Availability Rate, Hit Rate, and User Feedback Rate and thereby increases the request handles on the multiple users quickly by 14 – 43 % when compared to the DRR-CCMN [1]. Moreover, with the severance of the trustworthy and non-trustworthy software collection, dynamically, based on the Hit Ratio, the request handles an increase on multiple users by 23-53 % when compared to CS-OpenStack.

b. Scenario 2: Resource Utilization Efficiency on Multiple Users

To maximize the resource utilization efficiency on multiple users, the processor speed, bandwidth, and latency to handle the wide-ranging traffic conditions on the multiple user requests are deliberated. In the experimental setup, the number of requests positioned ranges from 20 to 140. As illustrated in Figure 5, the IST-MRC mechanism measures resource utilization efficiency, which is measured in terms of percentage (%). The resource utilization efficiency, using the IST-MRC mechanism, offers equivalent values with the state-of-the-art methods. Besides, the resultant resource utilization provides the summation of the processor speed, bandwidth, and latency, and it is obtained using the equation given below.

$$RU = \sum_{i=1}^{n} P_i, RAM, Storage_i \quad (6)$$

The targeting results of resource utilization efficiency, using the IST-MRC mechanism, compared with the two state-of-the-art methods, [1] and [2], are presented in Figure 5 for visual comparison, based on the number of the requests placed. The proposed approach differs from DRR-CCMN [1] and CSOpenStack [2] since we have incorporated competence optimization method that competently allocates the resources to the multiple resources. It can be evaluated under different traffic conditions by improving resource utilization efficiency. This value is 7–14% when compared to DRR-CCMN. Besides, the diverse traffic conditions, based on the processing speed and RAM speed, further enable the resource utilization efficiency by 2–18 % when compared with the CS-OpenStack [2].

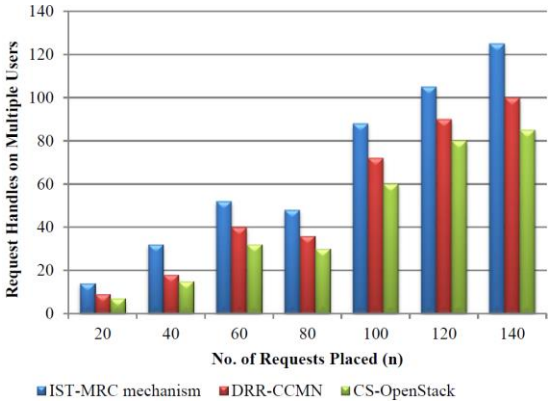

Figure 4. Distribution of request handles on multiple users in the experiments (n = 300 requests placed)

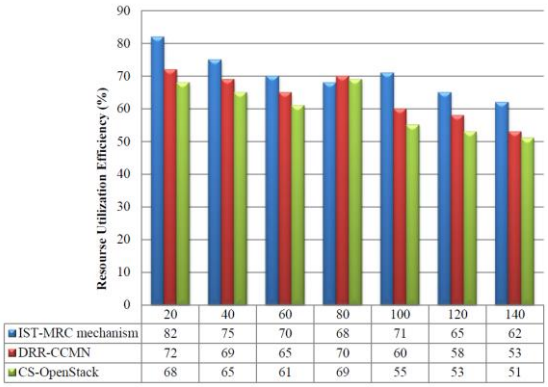

Figure 5. Distribution of resource utilization efficiency on multiple users in the experiments (n = 300 requests placed)

c. Scenario 3: Latency Time on Multiple Users

In Table 3, we show an analysis of Latency Time of IST-MRC mechanism concerning the behavior trust estimator value (BTE), ranging between 2.5 and 13.5, and measure the time taken to place a request between multiple users and the cloud server. It is calculated in terms of milliseconds (ms).

Figure 6 shows the Latency Time of the IST-MRC mechanism, DRR-CCMN [1], and CS-OpenStack [2] versus the increasing number of BTEs, from BTE = 2.5 to BTE = 13.5. The Latency Time improvement returned over DRR-CCMN, and CS-OpenStack decreases gradually as the number of BTE gets increased. For example, if BTE = 9, the growth of the IST-MRC mechanism, compared to DRR-CCMN, is 2.02 percent and when compared to CS-OpenStack, it is 4.92 percent. At the same time, if BTE = 12, the improvements are in the region of 1.64 and 3.56 percent, compared to DRR-CCMN and OpenStack, respectively. The trust value and resources are optimized separately in the IST-MRC mechanism. The proposed IST-MRC mechanism improves the Latency Time by 2 -7 % when compared to DRR-CCMN, and by 3–9 % when compared to CS-OpenStack.

d. Scenario 4: Latency Time on Trust Success Ratio

Figure 7 illustrates the Trust Success Ratio of the

different users, User 1 to User 7, taken for experimental purposes. As shown in this figure, the percentage of Trust Success Ratio in the IST-MRC mechanism is higher than the other two methods. The reason is the highest number of requests being handled, using the proposed IST-MRC mechanism. In addition, the feedback application, obtained from the repository, improves the Trust Success Ratio by 3 – 12 % when compared to DRR-CCMN, and 7 - 20 % when compared to CS-OpenStack.

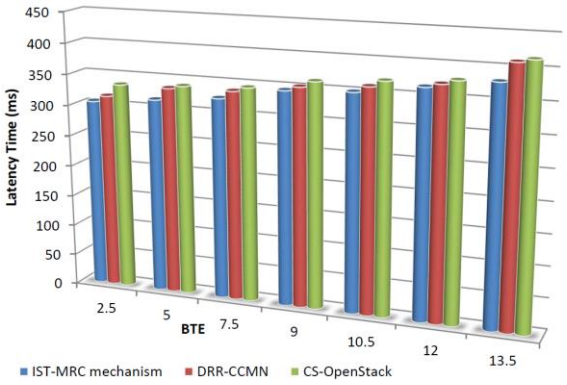

Figure 6. Distribution of Latency Time on multiple users in the experiments (BTE = 30 behavior trust estimators)

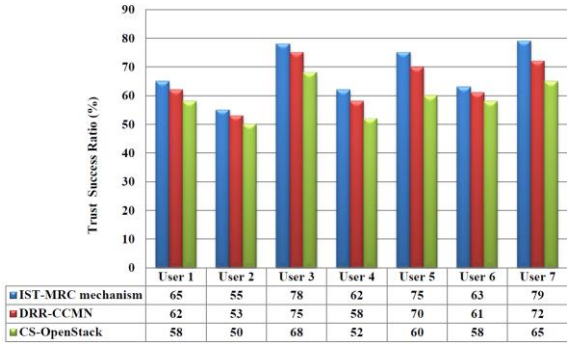

Figure 7. Distribution of Trust Success Ratio on multiple users in the experiments

TABLE 3. Tabulation of Latency Time

BTE	Latency Time (ms)		
	IST-MRC Mechanism	DRR-CCMN	CSOpenStack
2.5	305	318	335
5	315	325	340
7.5	325	338	345
9	345	349	352
10.5	350	353	360
12	365	370	375
13.5	380	395	410

6. CONCLUSION

In this paper, an integrated software trust estimator in alliance with the multi-user resource competence (IST-MRC) optimization mechanism was proposed in the cloud computing environment. The IST-MRC mechanism utilizes a behavior trust estimator to capably measure the trust value of the software service zone by applying Software Availability Rate, Hit Rate, and User Feedback according to the specific software apps. With the trust value computed for effective software provisioning to the multiple users, competence optimization is performed using processor speed, bandwidth, and latency to handle the varied traffic conditions on the multiple user requests. Finally, with the help of the feedback attained, the trustworthy and untrustworthy software collections are identified, and the trust value, based on the feedback, is provided. Experimental results reveal that the proposed IST-MRC mechanism not only leads to conspicuous improvement over Trust Success Ratio and Resource Utilization Efficiency but also outperforms the other methods - DRR-CCMN and CS-OpenStack- in Latency Time and successful requests handled on multiple users.

7. REFERENCES

1. Tang J., Peng Tay W., and Wen Y., "Dynamic request redirection and elastic service scaling in cloud-centric media networks." ***IEEE Transactions on Multimedia***, Vol. 16 (2014), 1434-1445. DOI: 10.1109/TMM.2014.2308726.
2. Abbadi I. M. and Ruan A., "Towards trustworthy resource scheduling in clouds." ***IEEE Transactions on Information Forensics and Security,*** Vol. 8, (2013), 973-984. DOI: 10.1109/TIFS.2013.2248726.
3. Wang Q., Wang C., Ren K., Lou W. and Li J., "Enabling public auditability and data dynamics for storage security in cloud computing." ***IEEE Transactions on Parallel and Distributed Systems,*** Vol. 22, (2010), 847-859. DOI: 10.1109/TPDS.2010.183.
4. Sundareswaran S., Squicciarini A., and Lin D., "Ensuring distributed accountability for data sharing in the cloud." ***IEEE Transactions on Dependable and Secure Computing,*** Vol. 9, (2012), 556-568, DOI: 10.1109/TDSC.2012.26.
5. Abbadi I, M. and Martin A., "Trust in the Cloud." Information security technical report, Vol. 16, (2011), 108-114.
6. Pandey S., Voorsluys W., Niu S., Khandoker A., and Buyya R., "An autonomic cloud environment for hosting ECG data analysis services." ***Future Generation Computer Systems,*** Vol. 28, (2012), 147-154, DOI: 10.1016/j.future.2011.04.022.
7. Takabi H., Joshi J. B., and Ahn G. "Security and privacy challenges in cloud computing environments." ***IEEE Security & Privacy,*** Vol. 8, (2010), 24-31, DOI: 10.1109/MSP.2010.186.
8. Wu X., Zhang R., Zeng B., and Zhou S., "A trust evaluation model for cloud computing." ***Procedia Computer Science*** Vol. 17 (2013), 1170-1177, DOI: 10.1016/j.procs.2013.05.149.
9. Sun D., Chang G., Sun L., and Wang X. "Surveying and analyzing security, privacy and trust issues in cloud computing environments." ***Procedia Engineering,*** Vol. 15, (2011), 2852-2856, DOI: 10.1016/j.proeng.2011.08.537.

10. Gao K., Wang Q., and Xi L. "Reduct algorithm based execution times prediction in knowledge discovery cloud computing environment." ***The International Arab Journal of Information Technology,*** Vol. 11, (2014), 268-275, DOI: 10.5296/npa.v7i2.7797.
11. Whaiduzzaman M., Nazmul Haque M., Chowdhury M. R. K., and Gani A. "A study on strategic provisioning of cloud computing services." ***The Scientific World Journal*** Vol. 14 (2014), DOI: 10.1155/2014/894362.
12. Wu Q., Zhang X., Zhang M., Lou Y., Zheng R., and Wei W. "Reputation revision method for selecting cloud services based on prior knowledge and a market mechanism." ***The Scientific World Journal,*** Vol. 4, (2014), DOI: 10.1155/2014/617087.
13. Sun Y., Zhang J., Xiong Y., and Zhu G. "Data security and privacy in cloud computing." ***International Journal of Distributed Sensor Networks,*** Vol. 10, (2014), 19-27, DOI: 10.1155/2014/190903.
14. Xiong A. and Xu C. "Energy efficient multi resource allocation of virtual machine based on PSO in cloud data center." ***Mathematical Problems in Engineering,*** Vol. 2, (2014), DOI: 10.1155/2014/816518.
15. Ghazizadeh E., Zamani M., Ab Manan J., and Alizadeh M. "Trusted computing strengthens cloud authentication." ***The Scientific World Journal,*** Vol. 4, (2014), DOI: 10.1155/2014/260187.
16. Varghese A. B., Hemalatha T, Sasidharan S., and Jophin S. "Trust Assessment Policy Manager in Cloud Computing-Cloud Service Provider's Perspective." ***International Journal on Recent Trends in Engineering & Technology,*** Vol. 10, (2014), 46, DOI: 10.5772/intechopen.76338.
17. Arianyan E., Taheri H., and Khoshdel V. "Novel fuzzy multi objective DVFS aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers*." **Journal of Network and Computer Applications,*** Vol. 78, (2017), 43-61, DOI: 10.1016/j.jnca.2016.09.016.
18. Wang X., Chen X., Yuen C., Wu W., Zhang M., and Zhan C. "Delay-cost tradeoff for virtual machine migration in cloud data centers." ***Journal of Network and Computer Applications,*** Vol. 78, (2017), 62-72, DOI: 10.1016/j.jnca.2016.11.003.
19. Kumar G. and Kumar Rai M. "An energy efficient and optimized load balanced localization method using CDS with one-hop neighbourhood and genetic algorithm in WSNs." ***Journal of Network and Computer Applications,*** Vol. 78, (2017), 73-82, DOI: 10.1016/j.jnca.2016.11.013.
20. Lin Y., Lai Y., Teng H., Liao C., and Kao Y. "Scalable multicasting with multiple shared trees in software defined networking." ***Journal of Network and Computer Applications,*** Vol. 78, (2017), 125-133, DOI: 10.1016/j.jnca.2016.11.014.
21. Chauhan S. S., Pilli E. S., Joshi R., Singh G., and Govil M. "Brokering in interconnected cloud computing environments: A survey." ***Journal of Parallel and Distributed Computing,*** Vol. 133, (2019), 193-209, DOI: 10.1016/j.jpdc.2018.08.001.
22. Xie X., Tianwei Y., Xiao Z. and Xiaochun C. "Research on trust model in container-based cloud service." ***Computers, Materials and Continua,*** Vol. 56, No. 2, (2018), 273-283, DOI: 10.3970/cmc.2018.03587.
23. Asadi F., and Hamidi H. "An architecture for security and protection of big data." ***International Journal of Engineering,*** Vol. 30, No. 10, (2017), 1479-1486, DOI: 10.5829/ije.2017.30.10a.08.
24. Xu X., Liu X., Xu Z., Wang C., Wan S. and Yang X. "Joint Optimization of Resource Utilization and Load Balance with Privacy Preservation for Edge Services in 5G Networks." ***Mobile Networks and Applications*** (2019), 1-12, DOI: 10.1007/s11036-019-01448-8.
25. Speily, B., Omid R., and Rezai H. "Energy aware resource management of cloud data centers." ***International Journal of Engineering,*** Vol. 30, No. 11, (2017), 1730-1739, DOI: 10.5829/ije.2017.30.11b.14.

Persian Abstract

چکیده

اعتماد سازی یکی از موارد مهم برای افزایش قابلیت مقیاس پذیری و قابلیت اطمینان منابع در محیط ابری است. در این مقاله به منظور ایجاد یک مدل مطمئن جدید بر روی منابع ابری SaaS (نرم افزار به عنوان سرویس دهنده) ، بهینه سازی استفاده از منابع برای درخواست های چند کاربره وتخمین اطمینان نرم افزاری به صورت یکپارچه با مکانیزم بهینه سازی منابع چند کاربره (IST-MRC) پیشنهاد شده است. مکانیسم بهینه سازی IST-MRC، بدون هیچگونه ترافیکی در محیط ابری، ویژگی های قابل اعتماد خود را در صورت نیاز برنامه های نرم افزاری، با تخصیص منابع بهینه ترکیب می کند. در ابتدا، تخمین اطمینان رفتاری در مکانیسم IST-MRC برای ارزیابی اطمینان سرویسهای نرم افزاری ایجاد می‌شود. ارزیابی اعتماد بر اساس نرخ در دسترس بودن نرم افزار، میزان نرخ ضربه و بازخورد کاربر در مورد برنامه‌های نرم افزاری خاص صورت می‌پذیرد. در مرحله بعد، منابع با استفاده از بهینه‌سازی شایستگی برای چندین کاربر بهینه می‌شوند. بهینه سازی شایستگی در مکانیسم IST-MRC سرعت، پهنای باند و تأخیر پردازنده را محاسبه می‌کند تا بتواند درخواست‌های کاربرهای مختلف را درشرایط مختلف ترافیکی مدیریت کند. آزمایش‌های مختلفی برای اندازه‌گیری و ارزیابی پارامترهایی از قبیل مدیریت درخواست موفق، بهره وری در استفاده از منابع، مدت زمان تاخیر و نسبت موفق اعتماد در چند کاربر انجام شده است.