ISSN 2515-8260 Volume 07, Issue 05, 2020

Multi-Attribute Performance Analysis Model For Collaborative Cloud Environment Using TCP

VelMurugeshKumar.N[1], R. Nedunchelian[2]

[1]Research Scholar, Department of Computer Science and Engineering, Saveetha School of Engineering,Saveetha Institute of Medical and Technical Sciences, Chennai, Tamilnadu, India

[2]Professor, Department of Electronics and Communication Engineering,Karpaga Vinayaga College of engineering and technology, Madhuranthagam, Chengalpattu District, Tamilnadu, India

Email: [1]velmurugeshkumar@saveetha.ac.in

Abstract: The problem of collaborative computing in the cloud environment has been well studied. Several approaches available to handle the issue of collaborative computing and still suffer to achieve higher performance in different QoS parameters. The performance of various simulation models has been analyzed for different parameters in this paper. To improve performance an efficient multi-attribute performance analysis model has been presented. The method first presents a collaborative content sharing strategy (CCSS) which maintains a dual copy of any resource in the cloud which has been accessed by different users. The method enforces an attribute-based update restriction (ABUR) algorithm which uses different keys to restrict malicious updating from different users. The public auditing has been enforced with ABUR and CCSS algorithms. The multi-attribute performance analysis model estimates various QoS parameters like throughput, completion, and public auditing on various simulation models of the cloud environment. A testbed is generated to simulate the performance of different cloud computing tools in a collaborative environment. The performance of simulation models has been measured and compared with the results of other models in detail.

Keywords: Cloud Computing, Collaborative Computing, Performance Analysis, QoS, Public Auditing.

1. INTRODUCTION

Modern organizations are growing over the ground of cloud computing. The organizations maintain a lot of information related to their customers and their own. However, it requires huge storage space, and purchasing such data servers costs higher and all the organizations are not capable of purchase costlier resources. Not only that, they require to use different resources which are more precious and cannot be afforded on their own. In this situation, the cloud has come to support the organizations. Cloud computing is a distributed paradigm that provides various services at different levels which can be accessed by different users of the environment.

The cloud environment allows the services to be accessed at the term of register and go. However, the services accessed are billed according to their usage. The organizations allow their users to access the resources from the cloud upon registration on different roles. This allows them to work over a single copy of data. The collaborative environment is the

ISSN 2515-8260 Volume 07, Issue 05, 2020

platform that makes it feasible for the employees of any organization to work together. Any organization would contain their units in different geographic locations of the world. However, they need to work together on a single instance of any resource. Whatever the update made on the resource should reflect on the copies of others who work on the same. The concept of collaborative computing makes this feasible.

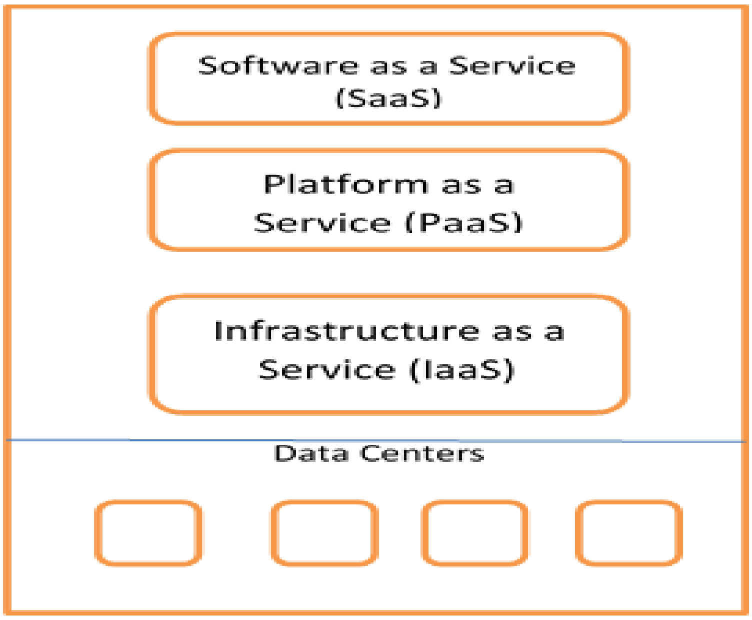

Figure 1 Service Level Architecture of Cloud Environment

Figure 1, shows the service level architecture of the cloud environment which has several layers and for each layer, a set of services are provided.

SaaS (Software as a Service):
The software services are provided in this layer. The software to access the cloud resource has been designed in a different language. But to access the cloud resource, the layer provides various services to compatible with any language.
PaaS (Platform as a Service):
The platform layer provides different services to handle the platform orient issues. The user may be using the service from a different platform and to compatible with any platform set of services that are provided in this layer.
IaaS (Infrastructure as a Service):
The infrastructure services are presented in this layer. The infrastructure issues are handled here to achieve higher service coordination.

Consider, a resource R of any organization x, present in location l, which has been accessed by N number of users who are located in the different geographic locations of the country. User A makes an update on the resource R which should be reflected on the resource of the same being accessed by the user B on the other location. The correctness and the originality of the data are more important in a collaborative environment. To achieve this, various protocols and methods have been discussed earlier, which restrict access to different levels and uses different authentication approaches. However, they suffer to achieve higher performance in public auditing.

On the other side, the throughput performance is another parameter to be considered in the problem of a collaborative environment. It is depending on the number of successful transactions performed in a collaborative environment which indirectly depends on the other

ISSN 2515-8260 Volume 07, Issue 05, 2020

parameters like latency. However, it is necessary to consider the different parameters of the quality of service of the environment. Similarly, completeness is the factor that affects the performance of the environment. It is measured based on the number of successful completion of update request generated from the users. By considering all the above-mentioned factors, the quality of service of the environment has been measured.

With this consideration, an efficient multi-attribute performance analysis model is presented which collaborative content sharing strategy and attribute-based update restriction algorithms in the cloud environment. The detailed approach is presented in the next section.

Simulation Tools Considered:

Several simulation tools available for the support of collaborative computing and the following tools are considered for performance analysis.

Cloud Sim:

A cloud sim is a tool that enables the operations of a cloud environment. The simulator enables the creation of multiple virtual machines on a single node and allows the scheduling of jobs and mapping them to a suitable virtual machine. The simulator supports the creation of multiple data centers and allows them to be accessed in an efficient way to support the scheduling of jobs [4].

Green Cloud:

The simulator presents the interface to simulate the energy-based protocols in a cloud environment with the support of Network Simulator 2. The method provides interfaces to monitor the energy depletion on different data center components and enables the data communication in form of packets to support load balancing [8].

Open Cloud:

This simulator provides various testbeds that are provided by a collaboration of Cisco, NASA, various universities of Japan, and the USA. The simulator provides various services in infrastructure, platform, and software layers.

Cloud Analyst:

The simulator supports the evaluation of large scale applications to monitor the behavior of various applications. The tool provides interfaces to distribute the services to be accessed globally. The tool allows the performance analysis of different applications with GUI components [5, 13].

Open Stack:

This is an operating system specially designed for the cloud environment which enables the controlling of various resources like storage, compute, and networking. All the resources can be controlled and simulated through web interfaces. The interfaces can be allowed to access through the amazon web services to enable the maintenance and retrieval of images.

2. RELATED WORKS

There areseveral algorithms discussed earlier for the problem of analyzing the performance of different cloud simulation platforms. This section details some of the methods related to the problem.

In [1], the author presents a detailed review of the simulation tools of the cloud environment. The author describes that the GDC tool is suitable for large scale application where the cloud analysis is suitable for handling data centers. The cloud sim has been identified as a tool to manipulate network operations where the hardware related problems are supported with MDC.

In [2], the problem of scheduling various services and applications in the cloud has been considered. The author performed a detailed study on how energy and load balancing can be performed. However, the author identified different simulators that support performance analysis. Similarly, in [3], the author performed a detailed review of various technologies that support cloud services and classifies cloud services based on user needs.

In [6], the author presented a detailed study on the performance analysis of security and other factors of quality of service. The analysis is performed due to the infeasibility of analyzing the performance using hardware as they cost higher. The analysis is performed through different simulation tools.

In [7], the author presents a prediction model that measures the performance of different cloud resources. The model has been evaluated the scaling utility of computing resources. The behaviors of the resources and systems have been captured and used to measure the performance of unknown metrics.

In [8], the author presents a comparative study on the performance of simulation tools like Opnenebula and Openstack. The analysis is performed according to the architecture, security, and provenance. Finally, the author provides different recommendations for the deployments based on the demands of users.

In [10], the author presented a large scale testbed towards the high-performance network services. The Open cloud Testbed adapted 4 data centers where each has 30 nodes. The data centers are located in Chicago, San Diego, and Baltimore. The testbed is widespread geographically which is connected with a 10Gb/s network which supports high-performance requirement.

In [11], a testbed named Open Cirrus is presented which provides data centers being distributed to support research and open-source service stacks in the cloud. Similarly, a testbed on the same which provides both physical and virtual machines. The services are distributed globally to perform monitoring and job submission.

In [14], the author presented a flexible infrastructure supportive simulator for a cloud environment named iCanCloud. The tool provides higher usability, flexibility, and more scalable. It also provides a user interface to perform various operations.

In [15], the author performs a detailed review of the cloud simulators and classifies them as pure software orient and software-hardware supportive. The features of each simulator class have been analyzed and discussed.

In [16], the author describes the features of the Cloud Sim simulator and compares its variants like green cloud, cloud analyst, EMUSIM, and MDCSIM. The author presents a comparative study on support towards platform services, network services, and language services.

In [17], the author presented a survey on cloud simulators according to mathematical models and different testbeds are used to measure their performance in various factors. The survey is performed to identify the suitable modeling type towards the evaluation of different cloud strategies.

In [18], the author presents an open-source framework to support cloud functionality named Eucalyptus to support control on the virtual machines. It is capable of controlling the virtual machines globally and the author outlined the architecture to support portability and simpler which can be deployed even in an academic environment.

In [19], the author presented a resource allocation strategy that has been simulated over CloudSim. As the simulator provides API to control the systems and behavior of the resources of cloud-like virtual machines. It supports the extension of any resource provisioning algorithm by modifying the existing ones.

In [20], the author simulated the infrastructure services and how they can be controlled through the Cloudsim simulator. The frameworks allow modeling and simulation of infrastructures with data centers. Also, allows the modeling of self-contained platforms to support scheduling and resource allocation policies.

In [21], the problem of environmental monitoring has been performed through cloud computing. The method developed an infrastructure to maintain the earth observation data. They designed a testbedthat allows the on-demand data discovery services which can be accessed by different scientists and researchers. The framework allows the integration of multiple source earth observation data obtained from different geographic locations and enables coordinative resource computing strategies.

In [22], a platform named HiTempo has been designed which supports the analysis of time-series data. The platform supports the analysis of satellite images and analyses various algorithms. The model-based algorithm like the Kalman filter has achieved a higher detection rate. Similarly in [23], the multi-source remote sensing data have been integrated to identify the changing conditions of land cover. The land cover maps have been used to perform the changing conditions.

In [24], the author a software architecture to support network-based applications. The author presents a detailed survey of various architectures that enable network services. Finally, a REST (Representational State Transfer) is presented and describes the guidelines to use through the modern web.

Enhancing integrated environmental modeling by designing resource-oriented interfaces [25], present design practices for creating resource-oriented interfaces, driven by an interaction protocol built on the combination of valid linkages to enhance resource integration, accompanied by associated recommendations for implementation. The suggested resource-oriented approach provides a solution to the problems identified above but still requires intense prototyping and experimentation. We discuss the central open issues and present a roadmap for future research.

The problem of earth monitoring has been observed in [26], which discusses the grid environment and its requirement to educate the earth observation a case study has been presented on earth observation and education.

In [27], a framework has been designed to handle the remote sensing images which provide interfaces to support the storage of images in Hadoop and retrieval, map-reduce, and so on.

ISSN 2515-8260 Volume 07, Issue 05, 2020

In [28], the author presented a review on spatial infrastructure in the cloud which analyzes the SDI and identifies the gaps present. The author presents a case study on SDI components to the value-added services.

In [29], the author presented an updated version of Multi-Verse Optimizer to achieve higher task planning in cloud. The author compared their proposed work with the existing algorithms and achieved minimized make span of time.

In [30], the author proposed an advanced cloud computing environment for library management system based on cloud. They have tested the performance of their system with various librarians, universities and proved that their design was secured and reliable.

In [31], the author presented a new complex system simulation method over the existing methods. This method provides a cloud environment service to perform simulations over the data.

In [32], the author has presented a network condition based low rate attack detection approach in multimedia networks which consider the network conditions like traffic, latency, number of routes available and other conditions in finding low rate attack. Similarly in [33] the author has presented an efficient Intrusion Tracking System using Region Based Traffic Impact Measure towards the Denial of Service (DoS) Attack Mitigation.

2.1 Multi-Attribute Performance Analysis Model:

The multi-attribute performance analysis model has been designed with several functional components to integrate the outcome of different cloud simulators. The method considers different parameters like throughput, completion, public auditing, number of services, number of API, backward compatibility, number of access, number of recommendations, and so on. The method enforces the access restriction according to the user profile at a different level. The entire data has been split into several categories and levels, each level of data has been restricted for a different profile. So, it has been verified for their trust according to the Collaborative content sharing strategy (CCSS) which estimates attribute level trust weight (ALTW) to restrict the user. On the other side, the dual authentication public auditing is performed which attributes level update restriction (ALUR) algorithm. The same has been adapted for different simulators of cloud and the performance has been measured. The detailed approach is presented below:

ISSN 2515-8260 Volume 07, Issue 05, 2020

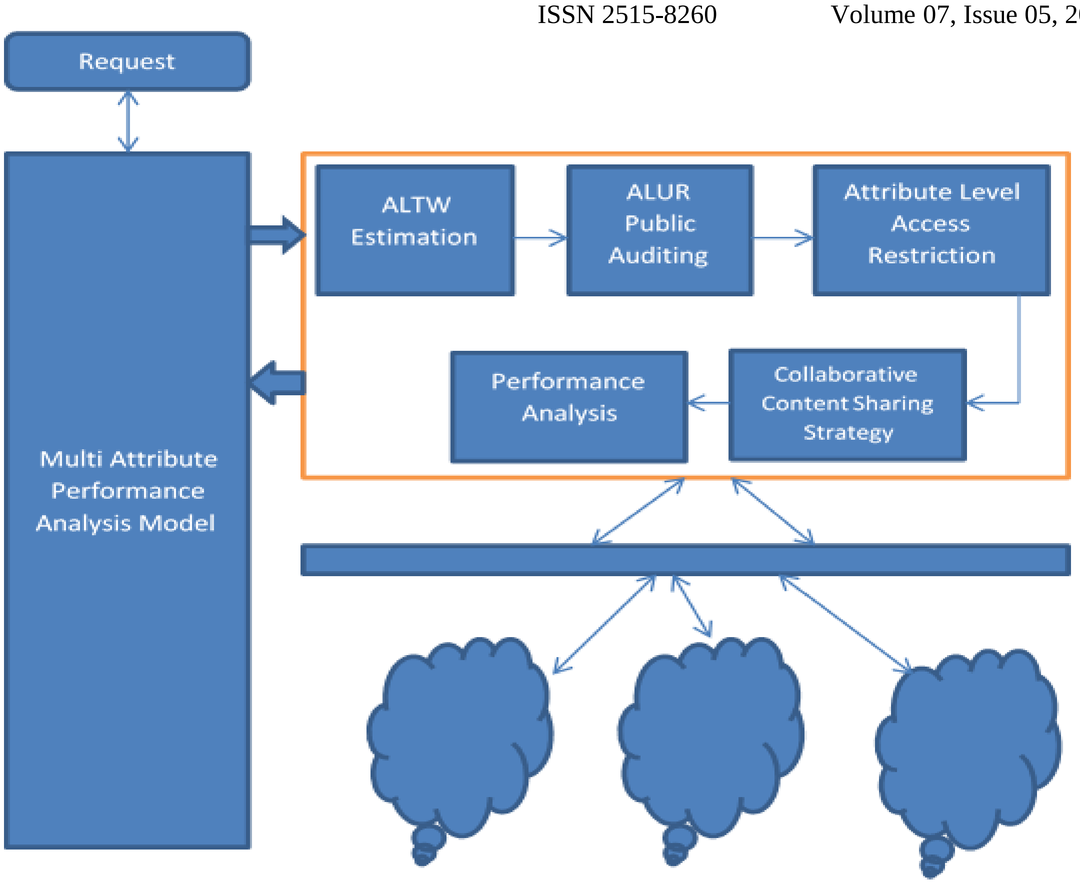

Figure 2 Block Diagram of Proposed Collaborative Cloud Environment

Figure 2, presents the block diagram of the proposed collaborative environment model and shows various components of the proposed system.

3. METHODOLOGY

3.1 Collaborative Content Sharing Strategy:

Public auditing has been identified as a key issue in a collaborative environment. To enforce higher public auditing in the environment, the method uses the content sharing model CCSS. The model first allows the data to be accessed with different users and the update restriction is performed with a different strategy. The method first performs access restriction according to the user profile and in attribute level. Second, the method estimates the trust weight and enforces public auditing to share the data with other users. Based on the trust weight being estimated, the method performs access restriction and performs public auditing to enable content sharing. Similarly, the method analyzes the performance of various simulation tools in achieving higher results.

Pseudo Code of CCSS:

Input: User request Ur, Data D, Taxonomy T.

Output: Null

Start

Read user request Ur, Data D, Taxonomy T.

While true

ISSN 2515-8260 Volume 07, Issue 05, 2020

Receive user request Ur.

If Ur.Type == Access then

Estimate ALTW.

If ALTW >Th then

Access the service and produce result to user.

End

Else

Estimate ALTW.

If ALTW >Th then

Result R = Perform ALUR Public Auditing.

Merge R with the User Content to produce result.

End

End

End

Stop

The above-discussed algorithm shows how the access restriction is performed and public auditing has been enforced in a collaborative environment.

3.2 Attribute Level Trust Weight Estimation:

The cloud data would have many details. For example, the medical data present in the cloud would have much information related to personal, diagnosis, and many. In a hospital, not all the users would not be allowed to access the entire information but each has been restricted according to their profile. Such an approach is presented in this paper, which restricts user access according to their profile. The method receives the user request and identifies the list of attributes the service request would access. From the profile of the user, the method would identify the list of the attribute to which the user has access. Using both of them, the method estimates the attribute level trust weight for each attribute level. The method classified the attributes into a different level and for each of them, the method would estimate the trust weight. Based on the trust weight at each level, the method estimates the cumulative trust weight. An estimated trust weight has been used to perform access restriction.

The lists of an attribute of any service request have been identified as follows:

$$\text{Attribute List Al} = \int_{i=1}^{size(AS)} \sum As(i) \rightarrow SR \quad \text{-- (1)}$$

Where AS is the attribute set and SR is the service request.

Similarly, the list of attributes to which the user has access has been identified as follows:

$$\text{Attribute Allowed for Access Aaa} = \int_{i=1}^{size(Al)} \sum Al(i) \in Uprofile(U) \quad \text{-- (2)}$$

Where U represents the user profile.

Based on these two values, the trust weight can be measured as described in the algorithm below:

ALTW Estimation Algorithm:

Input: Attribute Set AS, User Profile Uprofile, Service Request SR

Output: ALTW.

Start

Read user profile Uprofile, service request SR, Attribute Set As.

Al = Identify the Attributes List according to the Equation (1)

Identify the list of attributes of Al to which the user has access using equation (2).

Aaa = = $\int_{i=1}^{size(Al)} \sum Al(i) \in Uprofile(U)$ -- (2)

For each level L in attribute taxonomy AT

Compute ALTW = $\int_{i=1}^{Number\ of\ Levels} \frac{\sum Attributes(AT(l)) \in Aaa}{Number\ of\ attributes\ required\ in\ level\ i}$ -- (3)

End

Compute ALTW = $\frac{\sum_{i=1}^{Number\ of\ Level} ALTW(l)}{No\ of\ levels}$ -- (4)

Stop

The above-discussed algorithm reads the incoming request and identifies the list of attributes to be accessed from the cloud. Using the user profile and the taxonomy, the list of attributes to which the user granted has been identified. For each level of the taxonomy, the method estimates the trust weight to measure the ALTW measure. The trust weight measured has been used to perform access restriction.

3.3 ALUR Public Auditing:

In any collaborative environment, it is necessary to ensure the correctness of data being accessed. Consider the data instance D, which has been accessed by N number of users from different locations or organization which are globally distributed. To improve the performance of the cloud environment it is necessary to provide exact data to the user. It has been achieved by efficiently auditing the data. In our scheme, public auditing has been achieved by controlling the access and audit on multiple levels. As the data has been maintained in the number of levels, the public auditing is also performed at the attribute level. Whenever a request has been received for an update, the method identifies the list of attributes at a different level. For each level of attributes from the taxonomy, the method maintains a set of keys at each level of attribute taxonomy. The user can update only when they have the specific key for the attribute level. The method verifies the keys first and

ISSN 2515-8260 Volume 07, Issue 05, 2020

updates the view generated. The concurrency has been achieved by producing updates on the original at the update on view.

Algorithm:

Input: Update request Ur, Taxonomy T, User Profile Up, User Key Uk, Key Set Ks

Output: Null

Start

Read request Ur.

Identify list of attributes required to access ArA = $\int \sum Attributes(T) \rightarrow UR$

For each level l

Identify the list of attributes required Ar = $\int_{i=1}^{Nl} \sum Attr(ArA) \in l(i)$

Identify the no of attributes allowed Naa.

Naa = $\int_{i=1}^{Nl} \sum Attr(ArA) \in l(i)\ \&\&\ Up(User).l(i) == Grant$

Compute Level orient Update weight Louw.

LouW = $\frac{Naa}{Ar} \times 100$ – (5)

End

If LouW>Th then

If Uk$\in Ks$ then

Update in View.

Put lock on original data.

Generate update notification on all data instance.

End

End

Stop

The above discussed public auditing scheme uses a different set of keys for each level of attributes according to the taxonomy being used. At each level, the method estimates the level orient update weight to restrict the update process. If the user has a higher weight then he will be allowed to generate an update. Once the update is generated in the view of original data, then the update on the original has been performed with the notification to all the instances generated.

3.4 Attribute Level Access Restriction:

The users of the collaborative environment have been restricted according to the trust of the user in different stages. The method groups the attributes under different hierarchy according

ISSN 2515-8260 Volume 07, Issue 05, 2020

to their importance to the data. For each level, the method estimates the trust weight and finally, a cumulative ALTW weight has been measured. The trust weight has been measured according to the user's access grant on a different level of attributes. Based on the value of ALTW, the user has been restricted to access the cloud data in a collaborative environment.

3.5 Performance Analysis:

The proposed framework has been designed to measure the performance of a content sharing strategy with different simulation tools. For the evaluation, we have considered five different simulation tools. For each of them, different parameters like the number of API available, Number of services available, Popularity of tool, category, throughput, completion, and public auditing. Using these features, the method estimates the tool support measure (TSM) on each to produce the analysis results.

Algorithm:

Input: Content Sharing Strategy CSS, Batch Request Br, Taxonomy T, Simulators S

Output: performance Set Ps.

Start

Read CSS, Br, T, S.

For each simulator s

Create an instance I.

Initialize I with CSS, BR, T.

Start Simulation.

Compute No of Api available Naa = $\int_{i=1}^{size(S)} \sum Api(s(i)) \in S(s)$ – (6)

Compute no of services available Sa = $\sum Services \in s(i)$ – (7)

Compute Popularity Sp.

Sp=$\int_{i=1}^{size(Service\ Trace)} \sum Traces(i).Name == S(i) \&\& Traces(i).Status == Good$ -- (8)

Compute category NCat = $\sum cat\big(s(i)\big) \in ServiceTrace$

Compute throughput STh =$\frac{\int_{i=1}^{size(Service\ Trace)} \sum ServiceTrace(i).Name==S(i)}{size(Service\ Trace)}$ – (9)

Compute completion rate Scr.

Scr = $\frac{\int_{i=1}^{size(Service\ Trace)} \sum ServiceTrace(i).Name==S(i) \&\& Status==Complete}{size(Service\ Trace)}$ -- (10)

ISSN 2515-8260 Volume 07, Issue 05, 2020

Compute public auditing support Pas = $\frac{Number\ of\ correct\ data\ generated}{Total\ Requests}$ -- (11)

Compute Tool Support measures TSM.

TSM = $\frac{Naa}{STh} \times \frac{Sp \times NCat}{Sa} \times \frac{STh \times Scr}{Pas}$ -- (12).

Add to Ps.

End

Stop

The above-discussed algorithm represents how the analysis of various simulation tool's performance has been estimated. The method estimates the tool support measure for each tool according to various factors. Based on the support measure, the ranking has been performed.

4. RESULTS AND DISCUSSIONS

The proposed performance evaluation framework has been implemented using advanced java. The framework integrates various simulators with the content sharing scheme designed. The performance of various simulators has been measured for their performance in different parameters. The results obtained have been compared with the results of other methods. The details of the evaluation have been presented below.

Table 1: Evaluation Details

Parameters	**Value**
Simulators	Open Stack, Cloud Sim, Cloud Analyst, Green Cloud, Open Cloud
Number of request on Batch	200
Programming Tool	Advanced Java

Table 1, shows the details of the evaluation being used to measure the performance of various parameters. The methods have produced the following results.

Table 2: Simulators' properties

Simulator	**API Support**	**Number of versions**	**Language Support**	**Popularity**
Open Stack	Open Source	1	python	1.36
Cloud Sim	Open Source	3	Java	2.50
Cloud Analyst	Open Source	2	Java	0.23
Green Cloud	Open Source	1	C++/OTCL	1.79
Open Cloud	Paid	2	Hardware	1.35

Various properties of various simulators considered for evaluation has been presented in Table 2.

ISSN 2515-8260 Volume 07, Issue 05, 2020

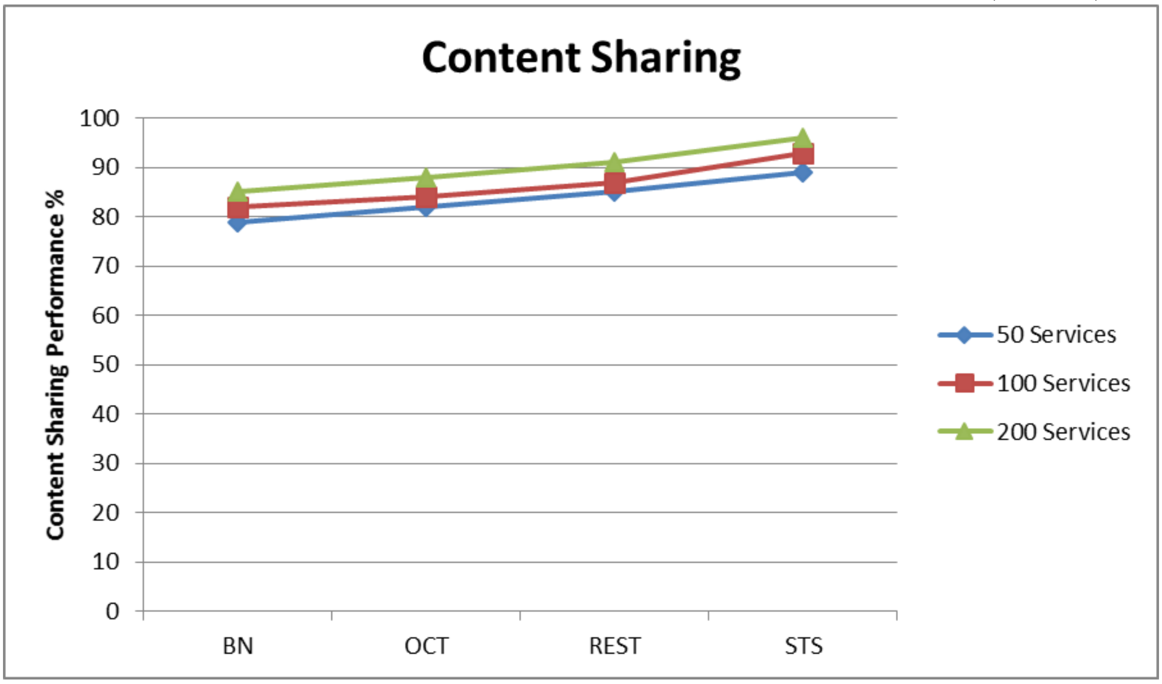

Figure 3: Performance of content sharing

The performance of content sharing has been measured and compared with the results of the proposed algorithm. The proposed STS algorithm has produced higher content sharing performance than other methods.

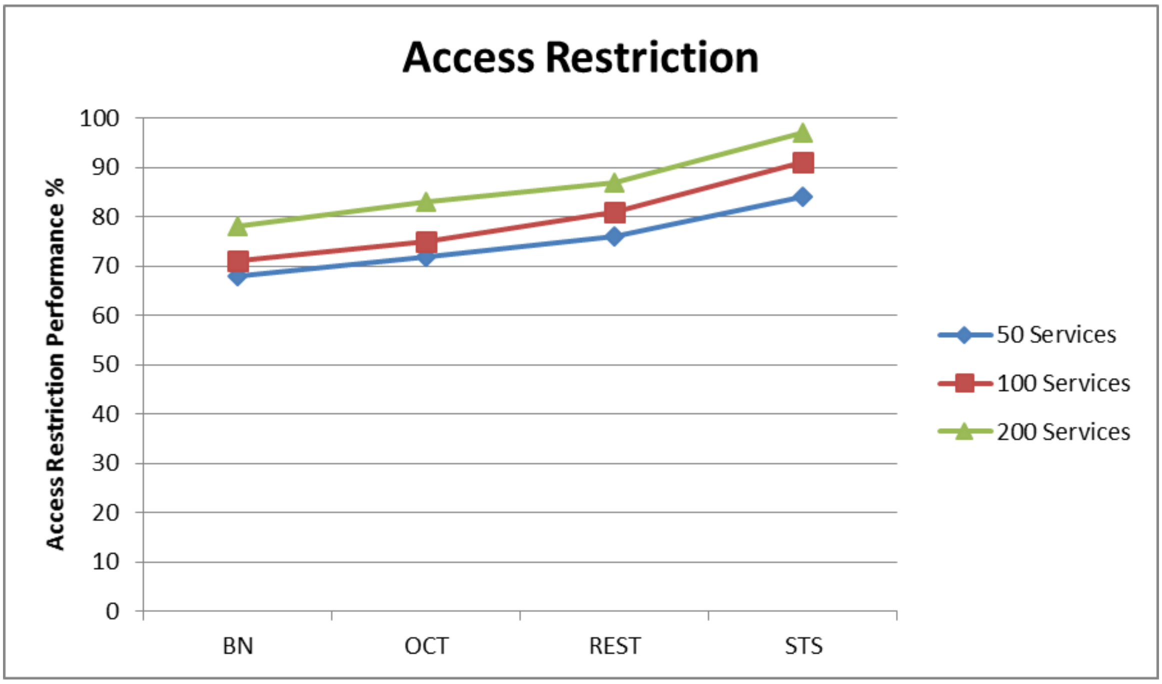

Figure 4: Access Restriction Performance

The performance of access restriction has been measured and compared with the results of other methods. The proposed STS algorithm has produced higher performance in access restriction than other methods considered.

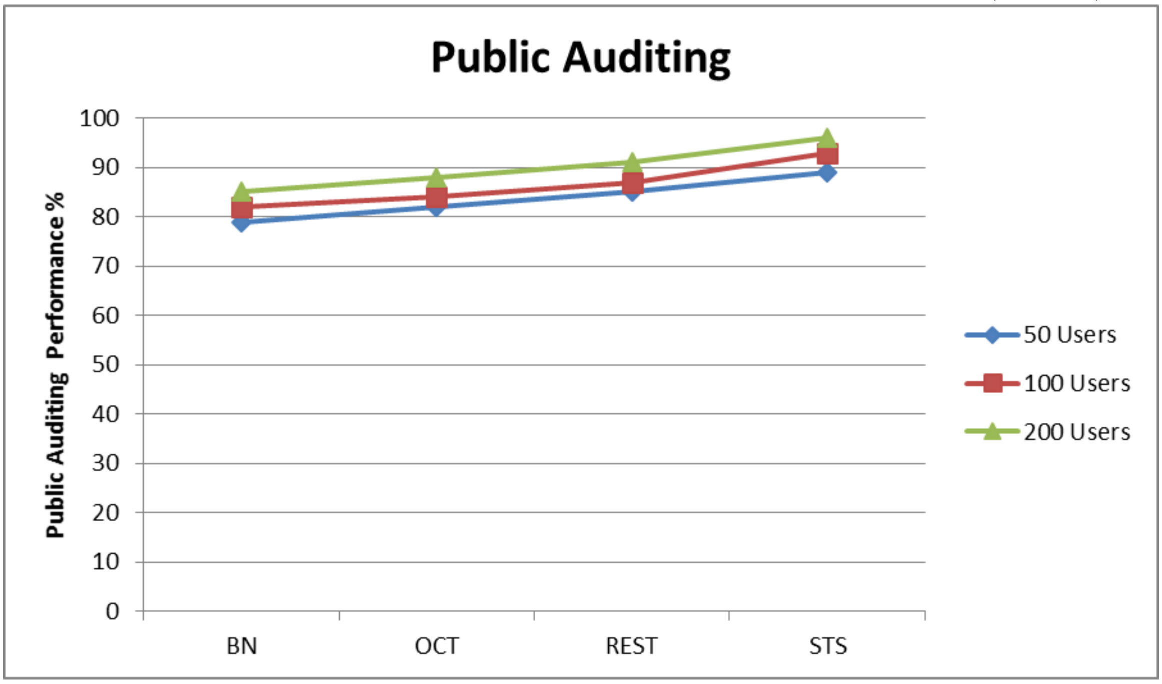

Figure 5: Performance on public Auditing

The performance on public auditing has been measured and compared with the results of other methods. The proposed STS algorithm has achieved higher public auditing performance than other methods.

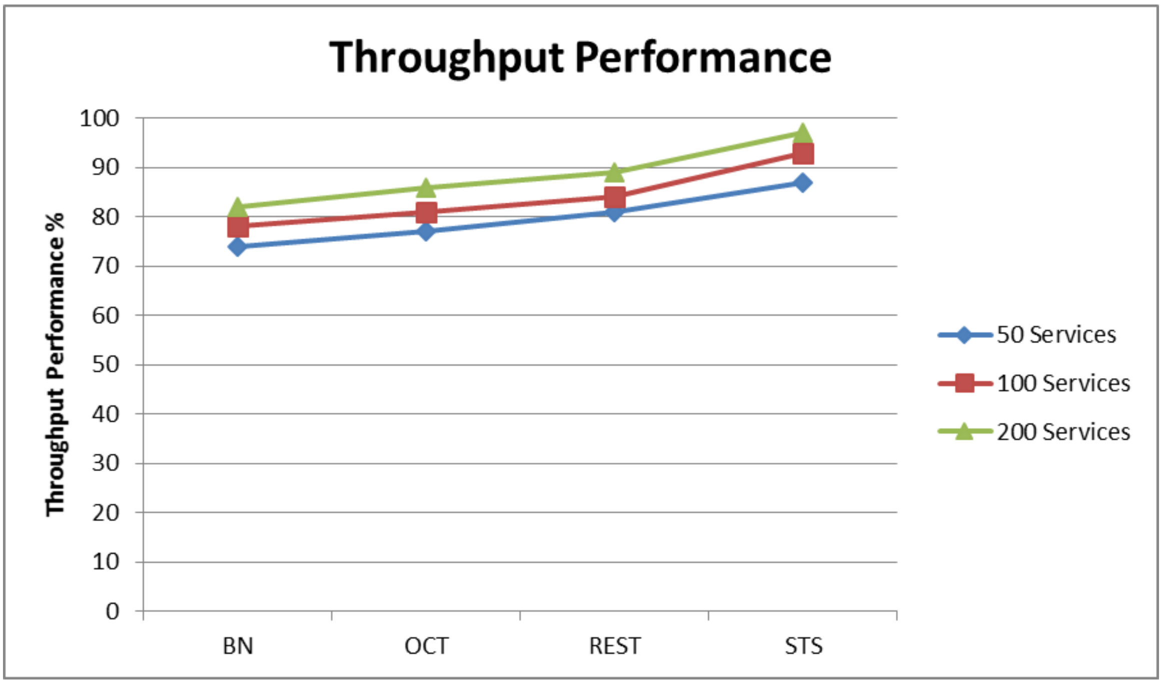

Figure 6: Throughput performance

The performance of throughput has been measured between different methods. The proposed STS algorithm has achieved higher throughput performance than other methods.

ISSN 2515-8260 Volume 07, Issue 05, 2020

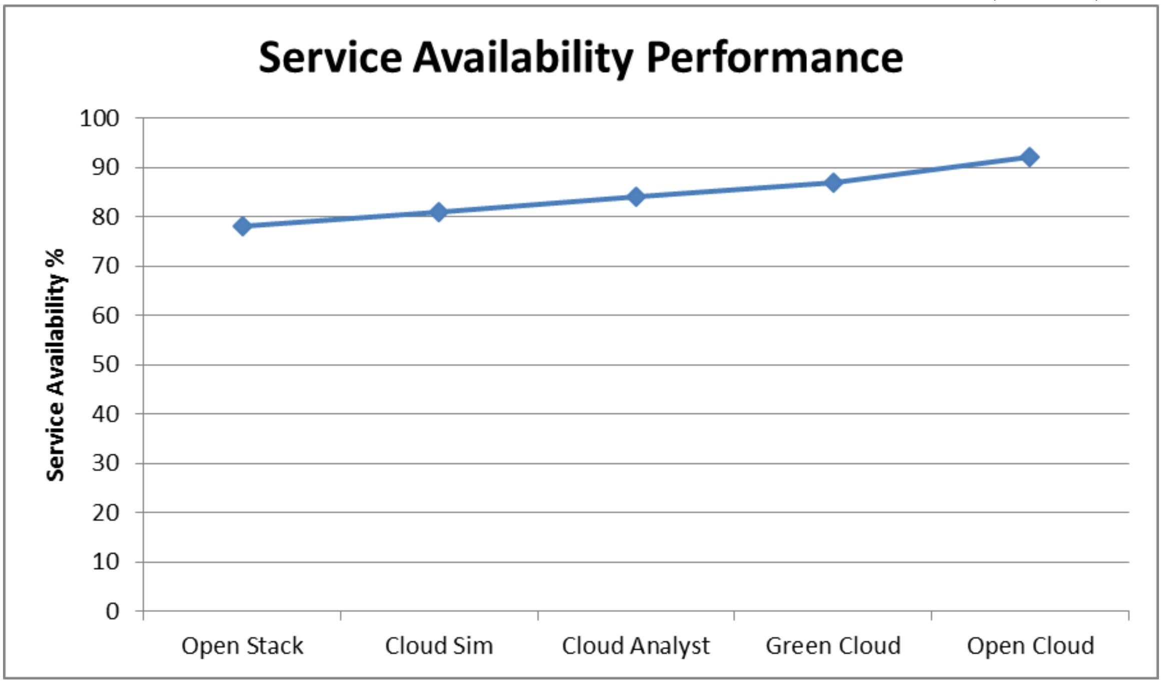

Figure 7: Performance of service availability

The service availability of any simulator is depending on the number of API it provides and the services it provides. The availability has been measured for all the simulators and presented in Figure 7.

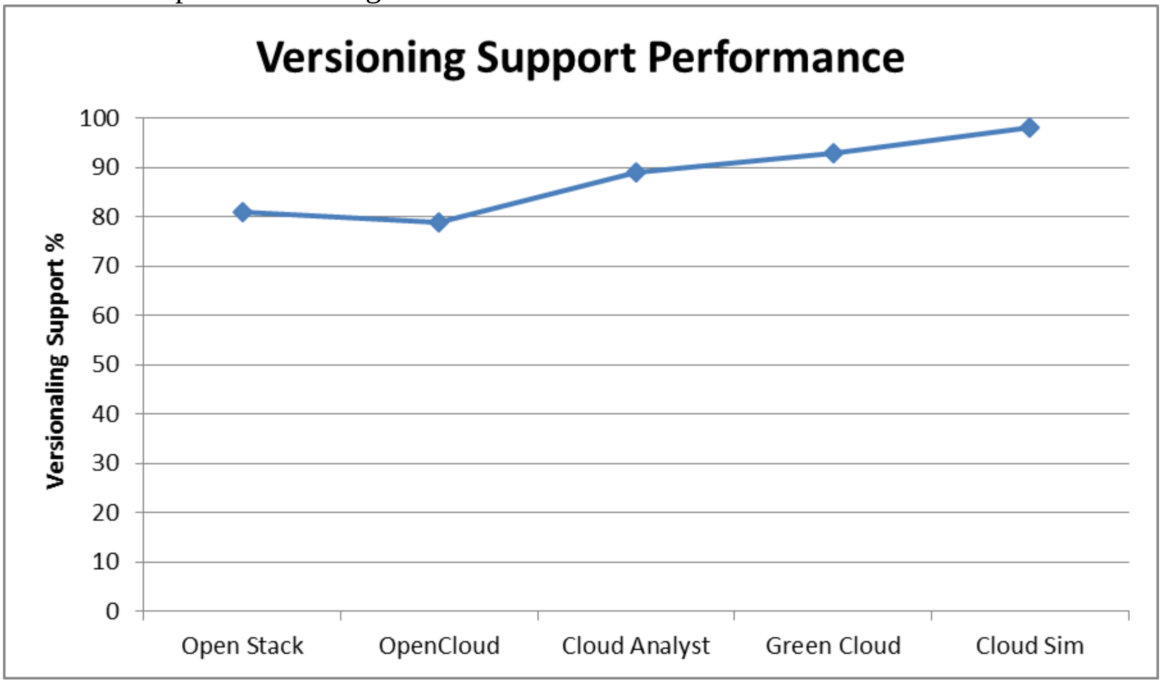

Figure 8: Performance of versioning support

The versioning support of different simulators has been measured and presented in Figure 9. It has been measured based on the number of versions of the simulator available for usage.

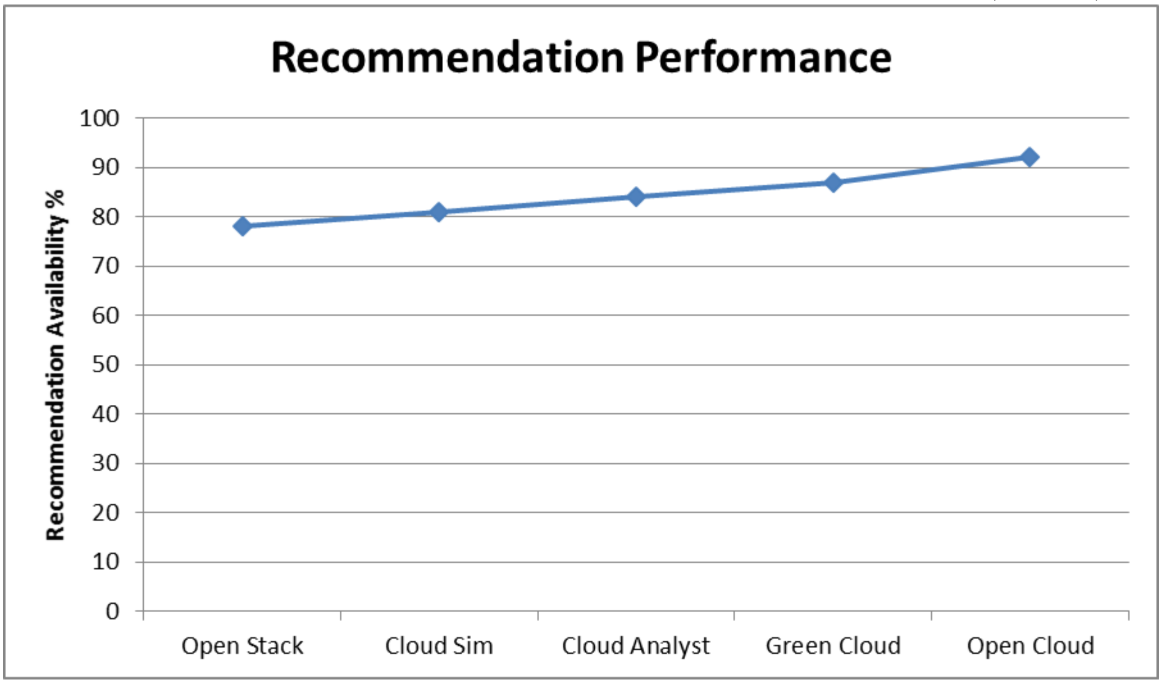

Figure 9: Performance of Recommendation

The recommendation performance on different methods has been measured and presented in Figure 9.

5. CONCLUSION

In this paper, an efficient multi-attribute performance analysis model has been presented. The framework has been designed to integrate some simulators based on the collaborative content sharing strategy. The content sharing strategy has been designed to perform access restriction according to attribute level access restriction which estimates the trust weight on each level. Similarly, the public auditing strategy has been designed to work based on the attribute-based update restriction (ABUR) algorithm which uses different keys to restrict malicious updating from different users. The public auditing has been enforced with ABUR and CCSS algorithms. The multi-attribute performance analysis model estimates various QoS parameters like throughput, completion, and public auditing on various simulation models of the cloud environment. A testbed is generated to simulate the performance of different cloud computing tools in a collaborative environment. The performance of simulation models has been measured and compared with the results of other models in detail.

Acknowledgments:
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

6. REFERENCES

[1] Rizwana Shaikh, M. Sasikumar, Cloud Simulation Tools: A Comprehensive Analysis, International Conference on Green Computing and Technology, 2013.
[2] Khairunnisa, M. NazreenBanu, Cloud Computing Simulation Tools: A study, International Journal of Fuzzy mathematic Archive, January,2015.

[3] RydhmBeri, VeerawaliBehal, Descriptive Study of Cloud Computing: An Emerging Technology, IJRITCC, March 2015

[4] Calheiros Rodrigo N, Ranjan Rajiv,"CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Wiley Press, New York, USA, January, 2011,pp. 23-50.

[5] B.Wickremasinghe (2009), "Cloud Analyst: A CloudSim based Tool for Modeling and Analysis of Large Scale CloudComputing Environments", MEDC Project Report, 2009.

[6] KaurGaganjot, Kumar Pawan,"Study of Comparison of Various Cloud Computing Simulators", 2nd National Conference in Intelligent Computing & Communication.

[7] Bashar, "Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models," in IEEE International Conference on Cloud Networking (CLOUDNET 2013), Nov. 2013, pp. 200–204.

[8] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, "GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Dat a Centers," in IEEE GLOBECOM 2010, Dec. 2010, pp. 1–5.

[9] X. Wen, G. Gu, Q. Li, Y. Gao, and X. Zhang, "Comparison of open-source cloud management platforms: Openstack and Opennebula," in IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), May 2012, pp. 2457–2461.

[10] R. L. Grossman, Y. Gu, M. Sabala, C. Bennet, J. Seidman, and J. Mambretti, "The Open Cloud Testbed: a wide-area testbed for cloud computing utilizing high-performance network services," in Gridnets 2009, Sep. 2009, pp. 89–97.

I. A. et al., "Open Cirrus: a global cloud computing testbed," IEEE Computer, vol. 43, no. 4, pp. 35–43, 2010.

[11] R. C. et al., "Open Cirrus Cloud Computing testbed: federated data centers for open source systems and services research," in USENIX Hotcloud '09, June 2009, pp. 1–5.

[12] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," in 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010) , Apr. 2010, pp. 446–452.

[13] Nez, J. L. Vzquez-Poletti, A. C. Caminero, G. G. Casta, J. Carretero, and I. M. Llorente, "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator," Journal of Grid Computing, vol. 10, no. 1, pp. 185–209, Mar. 2012.

[14] W. Zhao, Y. Peng, F. Xie, and Z. Dai, "Modeling and simulation of cloud computing: A review," in IEEE Asia Pacific Cloud Computing Congress (APCloudCC 2012), Nov. 2012, pp. 20–24.

[15] R. Malhotra and P. Jain, "Study and Comparison of CloudSim Simulators in the Cloud Computing," SIJ Transactions on Computer Science Engineering and its Applications (CSEA), vol. 1, no. 4, pp. 111–115, Sep. 2013.

[16] G. Sakellari and G. Loukas, "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing," Simulation Modelling Practice and Theory, vol. 39, no. 0, pp. 92–103, Dec. 2013.

[17] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus open-source cloud-computing system," in 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, May 2009, pp. 124–131.

[18] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, Jan 2011.

[19] R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, “Cloudsim: a novel framework for modeling and simulation of cloud computing infrastructure and services,” University of Melbourne, Tech. Rep., 2009.

[20] Wenyang Yu ; JiboXie ; Guoqing Li, Cloud computing for environmental monitoring using multi-source Earth Observation data, Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics), 2016.

[21] F. Van Den Bergh et al., "HiTempo: a platform for time-series analysis of remote-sensing satellite data in a high-performance computing environment", International Journal of Remote Sensing, vol. 33, no. 15, pp. 4720-4740, 2012.

[22] C. Petit, E. Lambin, "Integration of multi-source remote sensing data for land cover change detection", International Journal of Geographical Information Science, vol. 15, no. 8, pp. 785-803, 2001.

[23] R.T. Fielding, "Architectural styles and the design of network-based software architectures", University of California, 2000.

[24] C. Granell et al., "Enhancing integrated environmental modelling by designing resource-oriented interfaces", Environmental Modelling & Software, vol. 39, pp. 229-246, 2013.

[25] M. Neagul et al., "Web and Grid services for training in Earth observation", Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 2009. IDAACS 2009. IEEE International Workshop, 2009.

[26] F.-C. Lin et al., "The Framework of Cloud Computing Platform for Massive Remote Sensing Images", Advanced Information Networking and Applications (AINA) 2013 IEEE 27th International Conference, 2013.

[27] B. Schäffer, B. Baranski, T. Foerster, "Towards spatial data infrastructures in the clouds in Geospatial Thinking" , Springer, pp. 399-418, 2010.

[28] Sarah E. Shukri, Rizik Al-Sayyed, AmjadHudaib, SeyedaliMirjalili, “Enhanced multi-verse optimizer for task scheduling in cloud computing environments”,Expert Systems with Applications,2020,114230,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2020.114230.

[29] AdeyinkaTella, ScholasticaChizomaUkwoma, Adeniyi Isaiah Kayode, “A two models modification for determining cloud computing adoption for web-based services in academic libraries in Nigeria”,The Journal of Academic Librarianship,Volume 46, Issue 6,2020,102255,ISSN 0099-1333,https://doi.org/10.1016/j.acalib.2020.102255.

[30] Shuai Wang, Feng Zhu, Yiping Yao, Wenjie Tang, Yuhao Xiao, SiqiXiong, “A Computing Resources Prediction Approach Based on Ensemble Learning for Complex System Simulation in Cloud Environment”,Simulation Modelling Practice and Theory,2020,102202,ISSN 1569-190X,https://doi.org/10.1016/j.simpat.2020.102202.

[31] Suchithra, M., Baskar, M., Ramkumar, J. P. Kalyanasundaram, B. Amutha , “Invariant packet feature with network conditions for efficient low rate attack detection in multimedia networks for improved QoS”, J Ambient Intell Human Comput (2020). https://doi.org/10.1007/s12652-020-02056-1.

[32] Baskar.M, Gnansekaran.T “Developing Efficient Intrusion Tracking System using Region Based Traffic Impact Measure Towards the Denial of Service Attack Mitigation”, Journal of Computational and Theoretical Nanoscience, Volume No.14, Issue No.7, pp: 3576-3582, 2017.