AFTM-Agent Based Fault Tolerance Manager in Cloud Environment

Shivani Jaswal
University Institute of Computing
Chandigarh University
Punjab, India
shivanijaswal.uic@cumail.in

Manisha Malhotra
University Institute of Computing
Chandigarh University
Punjab, India
manisha.mca@cumail.in

Abstract: *As the number of cloud users are increasing with times, the probability of failures also increases that takes place in any cloud virtual machine. Failures can occur at any point of time in service delivery. There are numerous techniques for reacting proactively towards these failures. In this framework, a service provider is allocated to the user on the basis of ranking of the service provider. This ranking is done by considering parameters such as trust values (calculated by feedback mechanism), check pointing overheads, availability and throughput. Checkpoints are beneficial in triggering save point so that minimal loss of data takes place if any failure occurs. This paper has also compared the proposed framework with Optimal Checkpoints Interval (OCI) framework which is based on triggering checkpoints on constant rates. Results have proven that Agent based Fault Tolerance Manager (AFTM) has 33% to 50% better efficiency results as compared to OCI framework. The results shown in paper demonstrates how better the check pointing overheads, availability and throughput are handled by using AFTM framework. Also, the overheads were reduced to 50% as compared to OCI framework.*

Keywords: *Agents, checkpoints, virtualization, fault tolerant agent, overheads.*

Received January 6, 2021; accepted April 28, 2021
https://doi.org/10.34028/iajit/19/3/14

1. Introduction

Cloud Computing lies on the basis of virtualization. The various services are delivered to the users through Cloud Computing. These services can be software as a Service, Platform as a Service and Infrastructure as a Service. These cloud services can be delivered to its users through private, public, hybrid or community cloud [9, 24, 28]. The virtual machines are allocated to it users for the purpose of cloud services. These virtual resources have benefited small as well as large enterprise to migrate their data on Cloud [18].

Therefore, more the users, more is the probability that any fault can be arisen in the service delivery process. A fault is a state in which is able to work and is not as “unavailable” or “downtime”. Also, if a fault has been generated, service provider looks for other machines so that fault free services can be used by the consumer. A faulty system leads to non-fulfilment of tasks in given amount of time. The faults can occur in the system due to number of reasons such network failures, non-availability of resources, increasing workload [5, 29], system failures etc., [15].

Broadly, there are two techniques for handling fault tolerance i.e., reactive and proactive. In first technique, the damage to be caused by faults can be handled whereas in second technique, probability of fault occurrence can be reduced priorly. Out of both, the best method is of checkpointing that lies under reactive methods. In this a save point is being triggered in the System so that loss of data can be reduced to minimal even if the fault probability is there. This technique actually helps in dealing with other overheads that can lead to decreased efficiency of cloud service provider. In case of failure is recognised then the machine will restart its performance form last saved checkpoint.

In the case of large-scale enterprises, where high speed performances take place, sometimes, a system faces a failure which can be due to some execution constraints, addition or removal of multiple resources in the same cloud environment [6].

On the basis of techniques, number of fault tolerance models have been proposed in the past times. Many of them are able to check the faults but many times, overheads related to the technique has been noted. For example, in Optimal Check pointing Interval (OCI) i.e., checkpoints are used on constant rates. This leads to wastage of time, efficiency, checkpoints overheads. Therefore, a mechanism is required that can actually deliver a cloud service which is fault free and can predict the fault occurrence on the basis of history of service provider. This paper proposes a mechanism naming Agent based Fault Tolerance Manager (AFTM) that discovers the following:

- AFTM helps in identifying the fault occurring probability in environment and unfortunately, if fault occurs then minimal loss of data to be reported due to efficient triggering of checkpoints.

- The proposed mechanism works under various layers. Trust is also an important component here that contributes in identifying the cloud service providers those are actually ranked on the basis of various parameters.
- Similarly, fault is also one of the parameters that is considered for ranking of service provider so that user gets the best service provider and having minimal chance of fault occurrence.

This paper is an extension of AFTTM [19] in which there were layers which was Cloud Administrative Layer which interacts with Trust Evaluation layer for updation of trust values with trust agent. However, in AFTM, broker layer has been introduced that negotiates all the terms of SLA and provides services to the cloud users. Additionally, in AFTM, the checkpoints are triggered at one-third completion of the task. This helps in reducing the checkpoint overheads and hence reduces the monetary issues. Moreover, the main layer which is fault tolerance layer is overall handled by fault tolerance manager i.e., FTM.

Section 1 has covered the basic introduction of cloud computing followed by the underlining concepts of fault tolerance. Section 2 covers the related work. Section 3 proposed the framework. Section 4 illustrates the algorithm of mechanism in detail. Section 5 shows the evaluation results performed on Cloud Sim and its comparison with OCI. Last section i.e., 6 describes the conclusion and future scope of the proposed mechanism.

2. Related Work

Fault Tolerance has been considered as one of the most important issue in workflow management other than scheduling. In [4, 5] authors have used the technique of replication to handle faults in the system. This technique can be used in a system in which deadline is used for completion of tasks. In [27], authors have been shown appropriate balance between replication and resubmission techniques. But by following all these techniques, the performance is decreased and compromise increases in terms of Service Level Agreement (SLAs).

Mishra *et al*. [25], considered a mechanism for DDoS attacks mitigation based on reputation score policy and Bayesian game theory. In this, the knowledge based on probability concept is being utilized by cloud service provider to detect intrusion by malicious users within a cloud-environment.

Another technique that can be considered is check pointing which can be used other than replication and resubmission. In [14, 31, 36], authors have used check pointing technique that creates checkpoints periodically in between running tasks. The approach that has been proposed in [36], uses coordinated check pointing in two phases which actually increases overheads of the system. Nguyen and Desideri [26] uses the concept of independent check pointing which leads to domino effect.

Authors have proposed a secure framework using blockchain and key chain cryptography. It has actually helped in addressing the several issues related to data security and authentication in healthcare. Also, distributed framework to detect DDoS attacks in fog computing. Various parameters were considered such as rate of detection, rate of accuracy and false alarm rate. The proposed framework was far superior than the compared techniques. A framework was proposed namely Secured privacy preserving framework i.e., SP2F. It comprises of two engines i.e., two level privacy and deep learning-based engine. In this, SAE was used for converting data into encoded form of prevention of attack [21, 22, 23].

Author have proposed an algorithm that selected an individual fault tolerance technique for individual virtual machine. These techniques can be of replication method i.e., multi-version and parallel. A replication-based fault tolerance is proposed that actually reduced the service time and eventually increased the systems availability. Additionally, in this likelihood of forth coming faults is reduced. It is achieved by not allocating scheduled tasks to those servers whose rate of success is quite low [34, 35].

Akinwunmi *et al*. [1] have proposed an approach that identifies the trust worthy services with the help of several agents. Authors have performed experiments by considering response time and scalability. Several previous experiments were left out. Also fault tolerance concept can also be in uncalculated.

Hassan *et al*. [16] have proposed a Quality of Service (QoS) based trust model. In this accumulative value of trust is calculated and updated dynamically and is reflected each time to the providers. Also, a curative mathematical technique has been used to evaluate credibility of user’s feedback. Parameter of computing power of resources at run time is considered.

Srimachari and Anandharaj [30], presented a fault tolerance scheme that helped in reducing faults in a cloud environment by inducing coordinated checkpoints in virtual machine. This method helped in removing the unavailability status for checkpoint recovery.

Multi-Agent System (MAS) is a distributed system consisting of multiple software agents, which form “a loosely coupled network, called a MAS, to work together to solve problems that are beyond their individual capabilities or knowledge of each entity” [13]. MAS are a community of autonomous agents working together in order to achieve a goal [33]. Of particular interest are MAS in which the individual agents display significant intelligence and autonomy. Over the years, MAS technologies have found applications in many distributed systems such as distributed problem solving, distributed information fusion, and distributed scientific computing [12, 17, 20] .

Although there are many differences between cloud computing and MAS, both are two distributed computing models, therefore several common problems can be identified and many more benefits can be obtained by the combined use of cloud computing systems and multi-agents [31]. There are several researches that have attempted the use of agent technology in cloud computing.

Dahiya and Gupta [10], have proposed a technique based on mitigation that actually deals with network attacks. The technique possesses quite high detection rate of Distributed Denial of Service (DDoS) attacks and also the false positive rate is quite low. Authors performed a series of experiments by comparing three protocols. The final conclusion results in delay in miss rate, rate of restart and delay in communication [2].

In [3, 11], a paper was published in which authors illustrated a checkpoint method which is adaptive in nature. In this, the checkpoints those are not necessary are eliminated and additional checkpoints those are required are added in current cloud environment.

3. Proposed Framework

AFTM has works in multiple layers step by step. These layers are service consumer layer, broker layer, trust evaluation layer and virtualization layer embedded in service provider layer only as shown in Figure 1. A request is sent to broker through service consumer layer. In broker layer, there will be broker agent that will contact trust evaluation layer for getting the trust values of cloud service providers. Here, the trust values are calculated by the considering number of parameters such as availability, reliability, turnaround efficiency, response time, data integrity and fault tolerance. The broker agent will contact the concerned service providers available in-service provider layer having highest trust values for signing of agreements i.e., SLAs and negotiation of services. Finally, the service is delivered to the user and at last after the services usage, the feedback is taken by broker agent regarding the service usage. This feedback is submitted to trust evaluation layer contributing to dynamic and credible value of trust. Here, feedback is segregated as positive and negative feedback and furthermore, the values are computed along with parameters mentioned above so that trust value can be generated. Now, the virtualization layer is associated with service provider layer as shown in Figure 2.

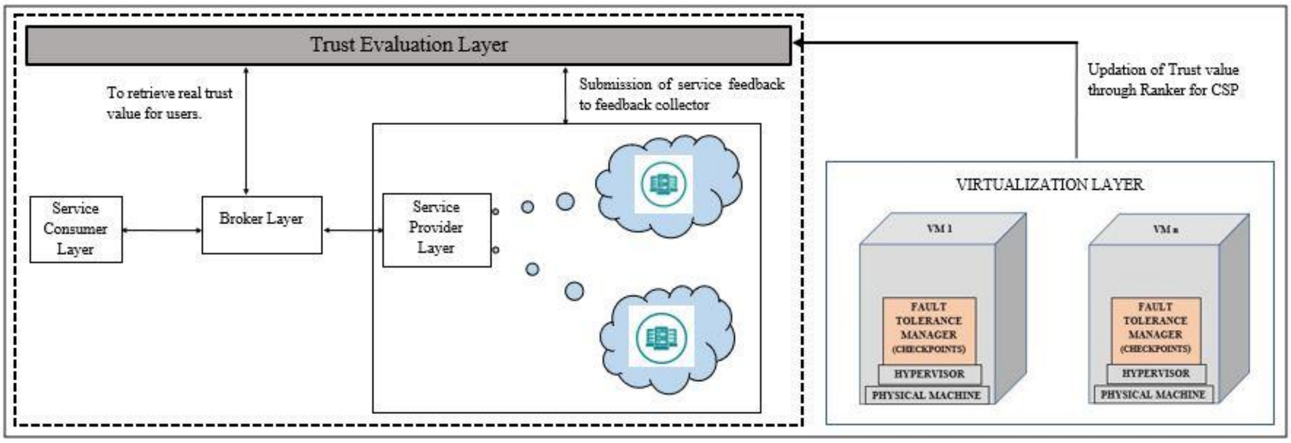
Trust Evaluation Layer
To retrieve real trust value for users.
Submission of service feedback to feedback collector
Updation of Trust value through Ranker for CSP
Service Consumer Layer
Broker Layer
Service Provider Layer
VIRTUALIZATION LAYER
VM 1
VM n
FAULT TOLERANCE MANAGER (CHECKPOINTS)
HYPERVISOR
PHYSICAL MACHINE
FAULT TOLERANCE MANAGER (CHECKPOINTS)
HYPERVISOR
PHYSICAL MACHINE

Figure 1. High level of AFTM.

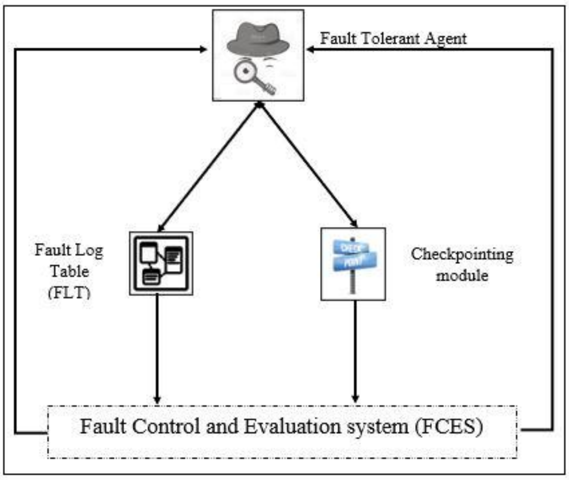

Figure 2. Detailed view of fault tolerance layer.

In this, a fault tolerance manager is installed over the hypervisor or virtual machine monitor which actually performs the principle tasks in this mechanism. Any type of faults occurring in the system is handled by this layer. In this layer, there are four modules working with each other.

- **Fault Log Table (FLT)**: This table consists of details regarding any fault occurring in system. It gets updated as long as the fault generates or removes. The structure of the data stored in FLT is shown in Table 1.
- **Fault Tolerant Agent**: This agent remains active throughout its lifecycle. It helps in communication among all the modules of this layer and at the end also submit the generated ranks to trust evaluation layer so that for next time, user gets the best service provider as available.
- **Checkpoint Module**: This module is based on reactive method of controlling fault tolerance. A FT agent is informed that a machine restart process is going to takes place and hence no further processing of the request will take place until and unless the reboot process in completed. The checkpoints are triggered considering history of service providers. If the history is good, then the lesser number of checkpoints will be used else vice versa. However, service provider having no previous history, then checkpoints are handled after every one-third completion of tasks.
- **Fault Control and Evaluation System (FCES)**: This is one of the most important module of fault tolerance layer. All the computations related to fault detection and evaluations takes place in this module.

Table 1. Format for FLT.

Host_ID	Fault_ID	Fault_Type	Status
H_01	F_01	VM Failure	Non - Active
H_02	F_05	Network unavailable	Non-Active
H_03	Null	Null	Active

4. Fault Tolerant Agent Algorithm

The fault tolerant agent works in virtualization layer in timely triggering of checkpoints and removal of faults (if any). The algorithm [19] depicts how the fault tolerant agent actually helps in smooth functioning of fault tolerance manager.

5. Performance Validation

Number of simulators are available for implementing simulation of cloud services. Cloud Sim is one of the most efficient simulators [7, 32]. It is easy to perform simulation on Cloud Sim and it is only simulator which can actually create the probability of occurring faults. In this, extra classes are created in which packages are imported. By creating these classes, new fault-based algorithms can be developed that actually monitors various virtual machines so that faults can be detected and resolved. Our proposed mechanism i.e., AFTM implements check pointing module.

The trust value has been evaluated by using following parameters as below:

Trust value $(\underline{T}i) = \sum_{i=1}^{n} \frac{Ti}{n}$

Where

$$Ti = \{(pi * \alpha) + (pi * \beta) + (pi * \sigma) + (pi * \mu) + (pi * \gamma)\} \quad (1)$$

Where *pi* represents are the feedbacks submitted by provider agent along with the parameters i.e., availability, reliability, data integrity, turnaround efficiency and response time and analysed by the feedback collector existing in the trust evaluation layer.

The proposed mechanism has been compared with OCI i.e., in this, the checkpoints are occurred at constant rates. The simulation results have been compared with OCI [8]. The parameters used in AFTM are throughput, availability and checkpoints overhead. In AFTM, the checkpoints are triggered as per the last ranking of cloud service provider. If the ranking is greater than threshold then the lesser number of checkpoints are used and vice versa.

Generally, for detection of failures event driven and time evolved techniques are considered. It works on the principle of stochastic process. In cloud system the random variables of time periods in following distribution and process considered in semi marked process.

In this model, it is assumed that Poisson distribution is followed. It denotes that the faults occurring is independent of the change of time. Therefore, following equation needs to be considered:

Failure probability distribution of VM in given time is given by:

$$Fp\ (Nn) = (e\text{^}(-\mu)\ \mu\text{^}n)/n!\ \ 0<Fp\ (N) <=1 \text{ and } n=0,\ 1,\ 2 \quad (2)$$

Where *N* (*n0*, *n1*, *n2*………) represents failures and μ represents the average number of failures.

The values of μ is given by:

$$\mu=fn/(Ti/\tau jn) \quad (3)$$

Where *fn*: number of failures and *Ti* represents the time period at which *fn* occurred.

τjn: Estimated time at which at which request occurred.

Probability of one error to take place is denoted by:

$$Fp\ (N1) = \mu e^{-\mu} \quad (4)$$

Finally, the rank of VM is calculated by considering another component known as ranker. Its value is obtained from status database. Hence, the equation is

$$Rp = \mu e^{-\mu}\ x\ Pi \quad (5)$$

Where *Pi* represents the percentage of profit earned through correct usage of VM

Table 2 shows the actual system configuration can which computations results were carried out in CloudSim.

Table 2. Values of configuration for AFTM computation.

List of Services	Assumed Values
Virtual Machines	10 (in number)
Ethernet Speed	100 Mbps
RAM	10GB
Storage	2TB
Customer Request	500 to 3500

6. Results

The results have been shown in this section and graph curves shows that the AFTM has better availability and throughput. Also, the checkpoint overheads have been reduced considerably as compared to OCI.

- **Case 1: Throughput**

In this, number of requests are shown in x-axis and y-axis depicts throughput results which is measured in requests per hour. Here, AFTM has been compared with OCI. Figure 3 shows that AFTM has better results as compared to OCI. It is because of the reason as before provisioning any service, its ranking is being considered through trust value that was generated. As, the ranking of the Cloud Service Provider (CSP) will be greater, more will be the throughput value of CSP and hence, better the results than compared one.

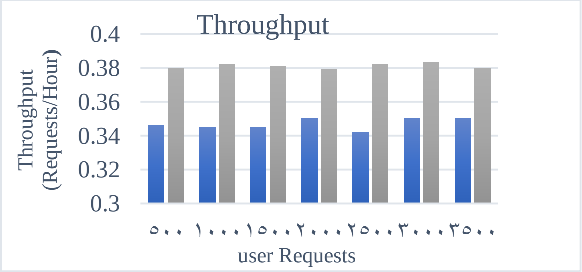

Figure 3. Comparison of AFTM and OCI on basis of throughput.

- **Case 2: Checkpoint overheads**

In this, user requests are represented on x-axis and throughput on y-axis as shown in Figure 4. The result graphs shows that checkpoints overheads have reduced considerably in case of AFTM. In this case, checkpoints are triggered as and when required. Therefore, unnecessary checkpoints will not be used and this has actually reduced the overheads in terms of checkpoints.

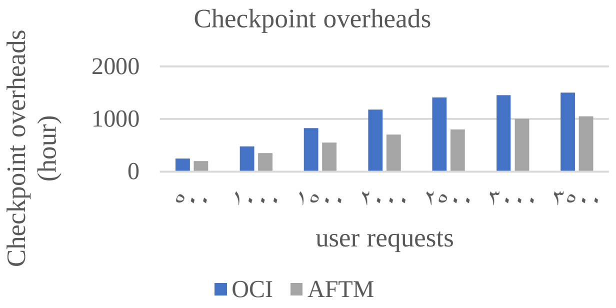

Figure 4. Comparison of AFTM and OCI on basis of checkpoints overheads.

- **Case 3: Availability**

As the cloud services are delivered on the basis of ranking and trust values. Therefore, availability of the concerned cloud service provider will be much higher as compared to OCI which has been shown in Figure 5.

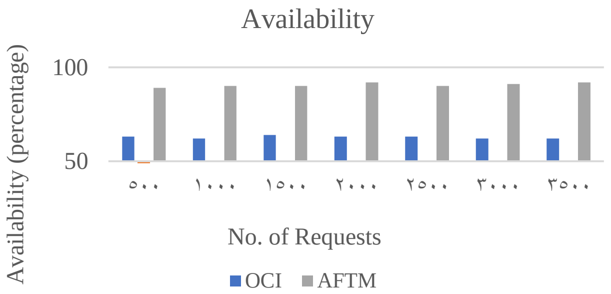

Figure 5. Comparison of AFTM and OCI on basis of checkpoints overheads.

7. Conclusions

As we know, failures can be avoided in cloud environment to some extent but not to full extend. To resolve this issue, a manager has been deputed in the virtualization layer naming AFTM: Agent based Fault Tolerance Manager in Cloud environment. This manager helps in generating checkpoints when and how required. It helps in inducing checkpoints and also reduces the loss of data that takes place during any fault. Many parameters have been used such as availability, checkpoints overheads and throughput. Results have proven that AFTM has better throughout and availability. The mechanism of checkpoint actually helps in reducing the loss of data caused due to faults in the delivery of services. Also, the overheads related to checkpoints have been reduced to half when AFTM is considered as unnecessary checkpoints will be avoided. The future work can be carried out on how other techniques of handling fault tolerance can be done so that faults can be managed timely with minimal loss of data. The further work can be carried out by considering restarting, replication techniques on AFTM so that efficient delivery of cloud services takes place.

References

[1] Akinwunmi A., Olajubu E., and Aderounmu G., "A Multi-Agent System Approach for Trustworthy Cloud Service Discovery," *Cogent Engineering*, vol. 3, no. 1, pp. 1256084, 2016.

[2] Al-Qerem A., Alauthman M., Almomani A., and Gupta B., "IoT Transaction Processing Through Cooperative Concurrency Control on Fog-Cloud Computing Environment," *Soft Computing*, vol. 24, no. 8, pp. 5695-5711, 2020.

[3] Amon M., "Adaptive Framework for Reliable Cloud Computing Environment," *IEEE Access*, vol. 4, pp. 9469-9478, 2016.

[4] Arockiam L. and Francis G., "FTM-A Middle Layer Architecture for Fault Tolerance in Cloud Computing," *IJCA Special Issue on Issues and Challenges in Networking, Intelligence and Computing Technologies*, vol. 2, pp. 12-16, 2012.

[5] Ben-Yehuda O., Schuster A., Sharov A., Silberstein M., and Iosup A., "Expert: Pareto-Efficient Task Replication on Grids and A Cloud," *in Proceedings IEEE 26th International Parallel and Distributed Processing Symposium*, Shanghai, pp. 167-178, 2012.

[6] Bilal K., Khalid O., Malik S., Khan M., Khan S., and Zomaya A., Fault Tolerance in the Cloud, *Encyclopedia of Cloud Computing*, pp. 291-300, 2016.

[7] Calheiros R., Ranjan R., Beloglazov A., De Rose C., and Buyya R., "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, 2011.

[8] Cao J., Simonin M., Cooperman G., and Morin C., "Checkpointing as a Service in Heterogeneous Cloud Environments," *in Proceedings of 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, pp. 61-70, 2015.

[9] Chen M., Ma Y., Song J., Lai C., and Hu B., "Smart Clothing: Connecting Human with Clouds and Big Data for Sustainable Health Monitoring," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 825-845, 2016.

[10] Dahiya A. and Gupta B., "A Reputation Score Policy and Bayesian Game Theory Based Incentivized Mechanism for DDOS Attacks Mitigation and Cyber Defense." *Future Generation Computer Systems*, vol. 117, pp. 193-204, 2021.

[11] Damodhar M. and Poojitha S., "An Adaptive Fault Reduction Scheme to Provide Reliable Cloud Computing Environment," *IOSR Journal of Computer Engineering*, vol. 19, no. 4, pp. 64-73, 2017.

[12] Drashansky T., Houstis E., Ramakrishnan N., and Rice J., "Networked Agents for Scientific Computing," *Communications of the ACM*, vol. 42, no. 3, pp. 48-ff, 1999.

[13] Durfee E. and Montgomery T., "MICE: A Flexible Test Bed for Intelligent Coordination Experiments," *in Proceedings of the Distributed AI Workshop*, pp. 25-40, 1989.

[14] Egwutuoha I., Chen S., Levy D., Selic B., and Calvo R., "A Proactive Fault Tolerance Approach to High Performance Computing (HPC) in the Cloud," *in Proceedings of 2nd International Conference on Cloud and Green Computing*, Xiangtan pp. 268-273, 2012.

[15] Gómez A., Carril L., Valin R., Mouriño J., and Cotelo C., "Fault-Tolerant Virtual Cluster Experiments on Federated Sites using BonFIRE," *Future Generation Computer Systems*, vol. 34, pp. 17-25, 2014.

[16] Hassan H., El-Desouky A., Ibrahim A., El-Kenawy E., and Arnous R., "Enhanced QoS-based Model for Trust Assessment in Cloud Computing Environment," *IEEE Access*, vol. 8, pp. 43752-43763, 2020.

[17] Honavar V., Miller L., and Wong J., "Distributed Knowledge Networks," *in Proceedings of IEEE Information Technology Conference, Information Environment for the Future (Cat. No. 98EX228)*, Syracuse, pp. 87-90, 1998.

[18] Jararweh Y., Alshara Z., Jarrah M., Kharbutli M., and Alsaleh M., "Teachcloud: a Cloud Computing Educational Toolkit," *International Journal of Cloud Computing* vol. 1, no. 2-3, pp. 237-257, 2013.

[19] Jaswal S. and Malhotra M., "AFTTM: Agent-Based Fault Tolerance Trust Mechanism in Cloud Environment," *International Journal of Cloud Applications and Computing*, vol. 12, no. 1, pp. 1-12, 2022.

[20] Khosla R. and Dillon T., "Intelligent Hybrid Multi-Agent Architecture for Engineering Complex Systems," *in Proceedings of International Conference on Neural Networks*, Houston, pp. 2449-2454, 1997.

[21] Kumar P., Kumar R., Gupta G., and Tripathi R., "A Distributed Framework for Detecting Ddos Attacks in Smart Contract-Based Blockchain-IoT Systems by Leveraging Fog Computing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, pp. e4112, 2021.

[22] Kumar R. and Tripathi R., *Blockchain Cybersecurity, Trust and Privacy*, Springer, 2020.

[23] Kumar R. and Tripathi R., "DBTP2SF: A Deep Blockchain-Based Trustworthy Privacy-Preserving Secured Framework in Industrial Internet of Things Systems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, 2021.

[24] Malik S. and Huet F., "Adaptive Fault Tolerance in Real Time Cloud Computing," *in Proceedings of IEEE World Congress on Services*, Washington, pp. 280-287, 2011.

[25] Mishra A., Gupta N., and Gupta B., "Defense Mechanisms Against DDoS Attack based on Entropy in SDN-Cloud Using POX Controller," *Telecommunication Systems*, vol. 77, no. 1, pp. 47-62, 2021.

[26] Nguyen T. and Desideri J., "Resilience Issues for Application Workflows on Clouds," *in Proceedings of ICNS2012-8th International Conference on Networking and Services*, Netherlands pp. 35-42, 2012.

[27] Palaniammal P. and Santhosh R., "Failure Prediction for Scalable Checkpoints in Scientific Workflows Using Replication and Resubmission Task in Cloud Computing," *International Journal of Science, Engineering and Technology Research*, vol. 2, no. 4, pp. 985-991, 2013.

[28] Pei X., Wang Y., Ma X., and Xu F., "Repairing Multiple Failures Adaptively with Erasure Codes In Distributed Storage Systems," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 5, pp. 1437-1461, 2016.

[29] Singh K., Smallen S., Tilak S., and Saul L., "Failure Analysis and Prediction for the CIPRES Science Gateway," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 7, pp. 1971-1981, 2016.

[30] Srimachari P. and Anandharaj G., "An Efficient Protocol Framework Solution for Resource-Constraint Mobile Devices Allocation in Cloud Computing Environments," *International Journal of Computer Science and Engineering Technology*, vol. 4, no. 4, pp.119-126, 2017.

[31] Talia D., "Cloud Computing and Software Agents: Towards Cloud Intelligent Services," *WOA*, vol. 11, pp. 2-6, 2011.

[32] Wickremasinghe B., Calheiros R., and Buyya R., "Cloudanalyst: A Cloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," *in Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications*, Perth, pp. 446-452, 2010.

[33] Wooldridge M., *an Introduction to Multiagent Systems*, John Wiley and Sons, 2009.

[34] Zhang M., Jin H., Shi X., and Wu S., "VirtCFT: A Transparent VM-Level Fault-Tolerant System for Virtual Clusters," *in Proceedings of IEEE 16th International Conference on Parallel and Distributed Systems*, Shanghai, pp. 147-154, 2010.

[35] Zhang Y., Zheng Z., and Lyu M., "BFTCloud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing," *IEEE 4th International Conference on Cloud Computing*, Washington, pp. 444-451, 2011.

[36] Zheng Z., Zhou T., Lyu M., and King I., "Component Ranking for Fault-Tolerant Cloud Applications," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 540-550, 2011.

Shivani Jaswal has an incredible record in teaching and education. She is pursuing her PhD from Chandigarh University in Cloud Computing. Her keen areas of research are Cloud Computing, Trust in Cloud Computing and Fault Tolerance in Cloud Computing. She has successfully published many papers in scopus and SCI journals. Also, she has published three chapters with scopus indexed. She is a member of various professional bodies such as ACM, IAENG etc. She is a reviewer of IJEBR, IGI Global.

Manisha Malhotra working as a Professor in Chandigarh University, India. She has credible record of various degrees like Ph.D (Computer Science & Applications), MCA (With Distinction), and BSC (Computer Science). She has published more than 20 research papers in various National/International Conferences, International Journal having indexed with Sci, Elsevier, Scopus, 34 and ACM. Dr. Malhotra is the members of various professional bodies like ACM, IEEE, CSI, and IAENG. She also has the members of editorial boards of various journals. She has been awarded as Young Faculty in the field of Cloud Computing in July 2016. She has also been awarded as Outstanding Researcher Award Green Thinker's Society in Interdisciplinary Research for Sustainable Development (IRSD – 2017) organized by Spoken Tutorial IIT Bombay, MHRD, Govt. of India at NITTTR, Chandigarh. Her research area includes Cloud Computing, Agent Technology, and Information Retrieval etc.