ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse281942020.pdf
https://doi.org/10.30534/ijatcse/2020/281942020

Fault Tolerance Analysis of Cloud Virtual Nodes in Real Time Applications

Mridula Dhingra[1], Dr. Neha Gupta[2]
[1]Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India
mridula.fca@mriu.edu.in
[2]Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India
neha.fca@mriu.edu.in

ABSTRACT

Cloud Computing provides the reliable services to clients through data centers which contains servers, storage etc. The major constraint of the systems used in real time applications is that, they are prone to failure. The failure may be due to the following reasons:

a) Failure to complete the task in prescribed time threshold value.

b) Failure to achieve prescribed reliability value.

To ensure reliability and availability of cloud technologies, methods to enhance fault tolerance needs to be developed and deployed. In the proposed work, Reliability Assessment Framework (RAF) has been developed to tolerate the faults of real time applications running on cloud infrastructure. The proposed RAF calculated the reliability of virtual nodes using adaptive fault tolerant method. Backward recovery technique has also been implemented to make the system fault tolerant and the calculated reliability value of virtual nodes help in replacing the failure nodes making the cloud platform more reliable and also enhanced the performance of cloud nodes.. Various parameters like Acceptance Computation, Minimum Reliability, Reliability Factor and System Reliability Level have been taken into consideration while implementing the framework.

Key words: Cloud Computing, Virtualization, Fault Tolerance, Reliability, Replication, Fine grained checkpointing, Reliability Assessment, Decision Mechanism.

1. INTRODUCTION

Cloud computing is one of the today's most thrilling concept for it's lessen expenditure related to computing while rising flexibility for computer processes [7]. In recent years, cloud has become a metaphor for voluminous data storage and utilization of virtual resources by cloud user [24]. Cloud computing is the current scenario is the most evolving technology due to its IaaS, PaaS and SaaS services [22]. It is an important computational standard which offers on-demand services to users and in low cost [10]. Cloud computing is used for manipulating, configuring, and accessing the hardware and software resources from distant regions [20]. It is important that the end user services across different applications need to be offered at one-stop [23]. Cloud computing alludes to the delivery of computing resources over the Internet [1]. Now a days, cloud computing is a developing field which offers reliable services to users worldwide. As the demand of cloud computing is increasing day by day, the cloud vendors should provide various services to users to meet their desired eminence needs. Cloud Computing is described as a model of computing resources such as storage, operating systems etc., in which virtualization is provided as a service over the Internet [12]. The benefits of Cloud Computing are low cost, pay-per-use on-demand-service, guaranteed QoS etc., but reliability is the major challenge among the users in cloud computing environment. Reliability means a particular task is accomplished within a particular time without failure. Reliability has the proficiency of a computer-related hardware or software elements to persistently accomplish the task coherent to its description.

The factors of reliability are throughput, response time, availability and serviceability. Throughput is the amount of data passing through a system or process in a specific time. Response time is the time to complete a particular process. To enhance the reliability, throughput should be high and response time should be less. Availability refers to the time taken by system to complete the task. Serviceability can be defined as the process through which any system can be maintained or repaired without much difficulty. In a cloud computing environment, when a system does not perform well as it is designed, then the system becomes less reliable or there is an occurrence of failure. A failure or fault is the condition when the hardware or software is unable to complete its functioning. As the computation in cloud is implemented on remote systems, so the probability of failures becomes high [14], so the fault tolerance capabilities are required to overcome the influence of system failures and to perform the task correctly when fault occurs.

1.1 Fault-Tolerance in Cloud Computing

Fault tolerance refers to correct and continuous operation of cloud nodes even in the presence of faulty components. It is the art and science of building computing systems that continue to operate satisfactorily in the presence of faults. A fault tolerant system may be able to tolerate one or more fault types [9] including- transient, intermittent or permanent hardware faults, software and design errors, operator errors, or externally induced upsets or physical damage. In real time cloud applications, processing on computing nodes is done remotely which has a high probability of occurrence of errors. These events increase the need for fault tolerance techniques to achieve reliability for the real time computing on cloud infrastructure.

The relationship between faults, error and failure is shown in figure 1. Each node is depended on each other. Occurrences of fault either in the component (Hardware) or in designing methodologies (Software) prone to error leads to the failure.

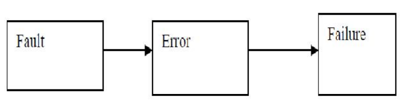

Figure 1: Creation trail of failure

a) A system is supposed to be fail when it does not accomplish the requirements of the users.

b) An error is the division of the system condition that can direct to failure.

c) The root of an error is a fault.

1.2 The Fault Tolerance Reliability Model

a) Fault tolerance provides the method or module to do the task effortlessly even though in the incidence of malfunction.

b) Fault tolerance assures quality of service for users by definite recital.

c) User can rearrange and append software instructions in accordance with their needs.

d) Fault tolerance is a technique or concept to implement a system to perform well in unexpected situations, e.g. when a part of cloud is not working up to the level of satisfaction the whole server of the cloud will not break down.

e) If the system is not entirely viable, fault tolerance results might permit a system to persist in service at compact ability rather than close down or abandon totally.

2. BACKGROUND OF STUDY

Attempts have been made in the literature to identify various techniques for fault tolerance in order to find the reliability of various nodes and systems in cloud computing for real time applications. Contribution of prominent authors are discussed as follows:

Authors [11] have presented various fault tolerance-related issues occurred in cloud computing. In their work, significant impressions and architectural facts related to fault tolerance were emphasized. The work reviewed existing fault tolerance techniques in cloud system that tolerates the central problems. The main intention of the work was to analyze the traditional fault tolerance approaches along with their respective challenges. Moreover effective solutions from the promising techniques were discussed that may lead to future research direction.

Authors [2] have discussed the fault tolerance system for cloud computing and evaluated their effectiveness. The authors have used two hardware based reliability modeling options that calculates the reliability of the system.

The authors [3] have discussed an adaptive scheme that handles the difficulty of fault tolerance in various cloud computing environments. The focus was on the adaptive behavior during the selection of replication and fine- grained check pointing methods for attaining a reliable cloud platform that can handle different client requests.

Authors [15] have proposed various fault tolerance and monitoring methods to increase the reliability of cloud computing infrastructure. The authors have described the data about various diverse techniques and methods used in fault tolerance and have also given prospect research guidance in cloud fault tolerance.

Authors [13] have proposed the Fault Tolerance Model for Cloud (FTMC).The model measures the reliability as the criteria for selection or rejection of nodes. The nodes producing incorrect results constantly have been rejected.

Various task re-submission and replication techniques have been used by the authors to tolerate the faults [16]. The replication factor of any task has been calculated by evaluating the replication and resubmission factors.

A resource allocation method has been discussed and authors have initiated a systematic representation which evaluates system reliability and resource restraints [17]. Memory and storage restraint of each server and highest load on each connection have been measured as the main resource restraints.

Authors [9] have introduced Fault Tolerance Management (FTM) approach based on layered abstraction method to develop and organize fault tolerance without execution facts. This method permits the clients to identify the preferred fault tolerance point to make it ready.

A model has been proposed with a status that permits merely reliable vendors who give the computing power and services that can offer a reliable environment for cloud computing. It has been discussed by the author that reliability of the operation is enhanced with the addition of diverse factors [19].

Fault tolerance method for real time computing on cloud infrastructure has been discussed by the authors [18]. The authors have proposed an Adaptive Fault Tolerance Model in Real Time Cloud Computing (AFTRC). This method uses forward recovery technique and is extremely fault tolerant. In this method adaptive reliability has also been used to increases the dynamic scalability of cloud environment.

Low Latency Fault Tolerance (LLFT) model has been introduced by the authors [21] using replication method that is implemented on distributed applications. This method reduces the faults of the applications by using various replication methods.

A lot of work has been carried out by different researchers in this field to find the reliability of virtual cloud computing nodes but with certain limitations. These limitations are:

a) Researchers have limited their scope to increasing reliability enhancement of virtual nodes, by -passing the issues related to cost-effectiveness and self recovery.

b) Extensive research has been done to reduce the failure intensity of nodes, however relationship with processing speed and dynamic scalability of cloud nodes have not been discussed much.

The above mentioned gaps motivated the author to enhance the reliability of virtual cloud nodes with respect to processing speed, cost-effectiveness and self-recovery.

3. PROPOSED RELIABILITY ASSESSMENT FRAMEWORK

A Reliability Assessment Framework (RAF) has been proposed to tolerate the faults of real time applications running on cloud infrastructure. The framework has been designed to tolerate the faults on the basis of reliability of computing node. In the proposed RAF, the failure of virtual machines is evaluated on the basis of following parameters:

a) Generation of Result.
b) Evaluation of turnaround time.
c) Verification of reliability parameters.

The Reliability Assessment Framework (RAF) will help in the following ways:-

1. Identification and replacement of failure nodes.
2. Implementation of backward recovery method to increase the reliability of the system.
3. Increase the dynamic scalability of cloud nodes using adaptive reliability method.

4. METHODOLOGY

A Reliability Assessment Framework (RAF) has been proposed to tolerate the faults of real time applications running on cloud infrastructure. The framework has been designed to tolerate the faults on the basis of reliability of computing node.

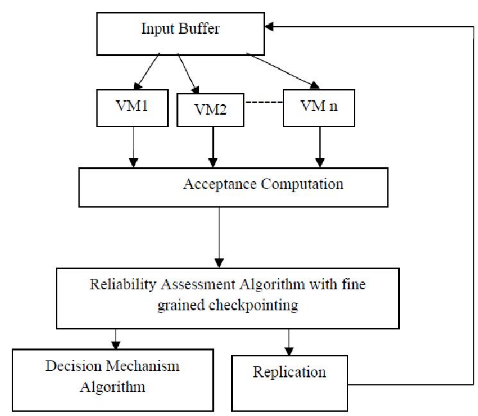

Figure 2: Proposed Reliability Assessment Framework (RAF)

The proposed framework in figure 2 consists of reliability assessment algorithm and decision mechanism algorithm as the core modules. RAF is used to access and enhance the reliability of virtual machines after every computation cycle. The framework has identified and replaced the failure nodes by computing minimum reliability level of the node and then implemented fine-grained checkpointing, if the node fails. If the node continued to fail, replication technique has been implemented to replace the node using backward recovery method. Decision mechanism module works simultaneously with reliability assessment to evaluate and enhance the reliability of the nodes. Decision mechanism selects the output of the node having the maximum reliability among all the competing nodes. Various parameters like Acceptance Computation, Minimum Reliability Level, Maximum Reliability Level, Reliability factor and SRL (System Reliability Level) have been taken into consideration while implementing the framework.

Every time the input is taken by the node, it performs the operation and provides the reliability value. This value has been passed to further modules for reliability calculation and decision making [4]. The proposed reliability assessment framework has been primarily divided into three modules, these are:.

A) Acceptance Computation

B) Reliability Assessment

C) Decision Mechanism

A) Acceptance Computation (AC) :- Each node has received the input from input buffer and produced the output. The output has been given to AC module. AC module is responsible for the verification of output generated by each virtual machine. Verification of output symbolizes the correctness of results. If the results are correct then AC module further calculates and verifies the threshold time required to complete the task assigned to virtual machine. If the results are found incorrect then backward recovery is performed through replication technique. If the results have exceeded the minimum threshold time then the node is considered to be faulty and backward recovery is performed through fine-grained checkpointing, otherwise the results are passed to reliability assessment module to measure the reliability of virtual machines and further to decision mechanism module.

B) Reliability Assessment (RA):- The reliability is a continuous measure and changes its value after every computation cycle. In this module, reliability assessment algorithm (proposed) is used to measure the reliability of virtual machines. After computation, if a virtual machine provides the correct results, i.e. the reliability value of virtual machine exists between minimum reliability and maximum reliability then output is forwarded to decision making module. However, if the reliability value of virtual machine is less than the minimum reliability then the backward recovery is performed by the system through fine-grained checkpointing and if the node continues to perform the incorrect results or gets failed, replication method is implemented for replacing the failure node [6]. In this module, three algorithms work simultaneously, one is reliability assessment algorithm to measure the reliability of virtual nodes and second one is fine-grained checkpointing algorithm for backward recovery and the replication algorithm for the replacement of failure nodes.

B1) Fine-grained checkpointing:- Most of the grid computing systems use check pointing as sensitive fault tolerance scheme to lessen the failure impacts [5]. The algorithm calculated the subsequent checkpointing interval at the time of the present checkpoint. It has been calculated by considering the VMs failure history on which the task is going to execute. In the situation of a poor failure history, the algorithm lessens the intervals between checkpoints. Additionally, the algorithm lengthens the checkpoint interval if there is a good failure history for a VM. The checkpoints are terminated at fixed interval which makes the cloud resources to be utilized that subsequently raises the checkpoint latency. Thus the objective of fine-grained checkpointing algorithm is to enhance the algorithm with the capability of determining the check pointing interval length adaptively premised on the VM's failure history. The checkpoint interval length assumed by our fine grained checkpoint algorithm is not fixed in the course of executing users requested job. The subsequent check pointing interval is only allowed to be calculated when executing the present checkpoint. This calculation is made concerning the failure history of virtual machine. The algorithm minimizes the checkpoint intervals once there is a poor failure history and lengthens it if there is a good checkpoint intervals.

B2) Replication:-Replication requires the minimum number of VMs should be available in cloud framework to handle the user's request. It becomes a costly affair to replicate each VM. So we only replicate those VMs which are executed and have larger impact on cloud performance if they are failed. Replication algorithm states that replica of every virtual machine is created and if the reliability time is less than the minimum virtual node development time and virtual node development time is less than the minimum creation time, then best virtual node has been evaluated, otherwise the existing virtual node is replaced with the new node.

C) Decision Mechanism (DM):- This module has been chosen as the concluding output for a computing cycle. DM module selects the output of the node with maximum reliability amongst all the computing nodes producing the correct result within threshold time. In DM module, system reliability level (SRL) has been predefined by the system. SRL is the minimum reliability level to be accomplished to pass the result. In this module decision mechanism algorithm (proposed) has been used to evaluate the best reliable node with the SRL and the best reliability must be greater than or equal to the SRL. If the node with reliability does not reach the system reliability level, the decision mechanism elevates the failure indication for that computing cycle and backward recovery is performed with the help of replication [5]. In this module two algorithms work simultaneously, one is decision mechanism algorithm to find out the best computing node on the basis of maximum reliability and the other one is replication algorithm for backward recovery.

In the proposed Reliability Assessment Framework (RAF), virtual nodes take input from input buffer and the failure of virtual machines has been evaluated on the basis of following parameters:

a) Generation of Result.

b) Evaluation of turnaround time.

c) Verification of reliability parameters.

In the proposed RAF, Cloudsim is used to simulate the cloud environment to test the reliability and to generate the results. Among the several accessible cloud simulator domains, Cloudsim is performing well. The classes of the package permit the development of algorithms based on fault tolerance which in turn can supervise virtual nodes towards the identification of failures and then later resolves them. This provision can deploy both fine grained check pointing and replication mechanisms. This simulator offers the capability to calculate availability, throughput, time overhead and monetary cost overhead [3].

5. IMPLEMENTATION OF RELIABILITY ASSESSMENT FRAMEWORK

The virtual machine's reliability is adaptive and is changing with each computing cycle. The reliability of every single machine is always perfect at the beginning stage. At the time of managing the processing node, its reliability increases to acquire correct result within less time. In case of failure of making a correct processing node, its reliability increases. In such a case a Reliability Assessment Algorithm (RAA) is utilized for measuring its minimum and maximum reliability level. The RAA halts the further working of node when there is a drop in the processing node below its minimum reliability. At that time the fine grained check pointing is used for recovering that node. However the replication technique has been used if the node reaches below the minimum reliability level more than three times. Afterwards, the replication technique is used which means the faulty node has been exchanged by the new node, this is done by backward recovery. On the other hand, if the reliability level of the node gets satisfied reaching above the minimum reliability it has been directed towards the decision mechanism module that picks out the final output for a computing cycle. Thus the selected output node is having the highest reliability among the other competing nodes that produced the results within shorter time. System reliability level has to be considered at minimum level to allow the result. The best reliability is compared with system reliability level via DM and it has to be either greater than or equal to system reliability level. In case of not achieving the SRL with best reliability node, the failure signal is raised for the computation cycle or else it gets permitted.

6. RESULTS AND DISCUSSIONS

The proposed reliability framework has been simulated on Cloudsim to check the execution time and reliability value of virtual machines. In the proposed RAF, the Initial Reliability (IR) value of a node is set to 1. Input parameters required in RAF are: Reliability Factor (RF), Maximum Reliability (MAR) and Minimum Reliability (MR). The values of RF, MAR and MR may vary depending on execution of real time applications. The minimum reliability level of the node is computed with this reliability assessment to find and replace the failure nodes. Reliability value of the node is calculated using

$$IR = IR + (IR * RF) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \text{ eq. 1}$$

Ten virtual machines have been taken into consideration while checking the reliability of nodes and on the basis of reliability value the virtual machine with the highest reliability will be selected. The following results have been achieved from the proposed framework.

Table 1: Measurement of Reliability Value using proposed RAF

VMID	Total time (ms.) (1000)	Reliability Value	Result	Best VM Selected
0	852	0.99	Pass	VM 7 with VMID 6
1	114	1.99	Pass	
2	366	2.99	Pass	
3	954	3.98	Pass	
4	776	4.98	Pass	
5	407	5.98	Pass	
6	747	6.98	Pass	
7	562	0.76	Fail	
8	134	0.79	Fail	
9	137	0.87	Pass	

Table 1 displays the results of 10 Virtual Machines (VMs). Out of 10 virtual machines, 8 machines have passed the reliability test and two have failed. Also VM7 with VMID6 is selected as best VM on the basis of its highest reliability value. Nodes having reliability value less than 0.8 are considered fail and the input of the failed VMs will be reallocated to the VMs having good reliability value.

The graphical representation of the table is represented in figure 3.

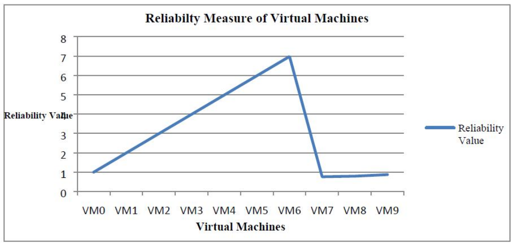

Figure 3: Reliability Measure of Virtual Machines

Table 2: Experimental Results of Proposed RAF

Cycle	Virtual Machine 1			Virtual Machine 2			Virtual Machine 3			Best VM Selected
	Time (Ms.)	Reliability Value	Result	Time (Ms.)	Reliability Value	Result	Time (Ms.)	Reliability Value	Result	
1	852	0.9973	Pass	114	1.994	Pass	366	2.992	Pass	VM3
2	840	0.9823	Pass	123	0.713	Fail	421	1.993	Pass	VM3
3	810	1.278	Pass	234	0.921	Pass	433	1.265	Pass	VM1
4	792	1.326	Pass	233	0.678	Fail	562	2.812	Pass	VM3
5	780	2.678	Pass	321	1.896	Pass	561	2.214	Pass	VM1

Table 2 displays the performance of three virtual machines for 5 cycles. Virtual machine 2 has failed in two cycles whereas virtual machine 1 and 3 has passed all the cycles. The checkpointing technique was then applied on virtual machine 2 resulting in it being considered as live node. If checkpointing technique is not applied, the machine facing the failure in any of the cycles is removed.

The result of proposed RAF has been compared with existing fault tolerance models on the basis of following parameters:

a) Average Execution Time

b) Average Reliability Value

c) Reliability Status of Nodes

Table 3: Comparative Analysis of Proposed RAF with Existing Fault Tolerance Models

Model Name	AFTRC (Sheheryar & Fabrice 2011	FTMC (Meshram, Sambare & Zade 2013)	High Adaptive Fault Tolerance in Real Time Cloud Computing (HAFTRC) (Rai, Kumar & Manisekaran 2014)	Adaptive Fault Reduction Scheme (AFRC) (Damodhar & Poojitha 2017)	Proposed RAF
Average Execution Time (Ms.)	0.78 (Threshold Time 2500 Ms.)	0.94 (Threshold Time 2000 Ms.)	0.74(Threshold Time 1800 Ms.)	0.91(Threshold Time 1500 Ms.)	0.50 (Threshold Time 1000 Ms.)
Average Reliability Value	1.82	1.69	2.42	2.69	3.599
Reliability Status of nodes in %	60%	60%	70%	70%	80%

Table 3 depicts the average execution time, average reliability value and reliability status of nodes of existing and proposed framework. The proposed framework accomplished better results with lesser average execution time and more reliability value while comparing with other methods. It is also clearly visible that number of failure nodes of proposed framework are also lesser than the existing nodes.

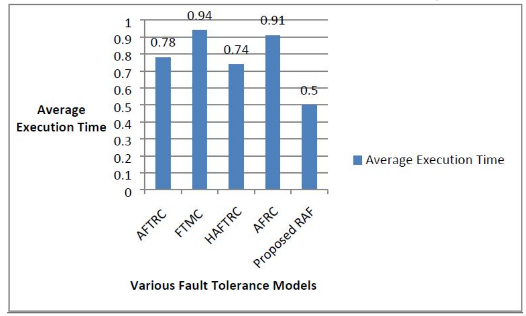

Figure 4: Comparison of Proposed RAF with Existing Models w.r.t their Average Execution Time

In figure 4, average execution time of proposed framework is compared with existing fault tolerance modes and the graph depicts that the proposed framework is having the least average execution time in comparison with existing models.

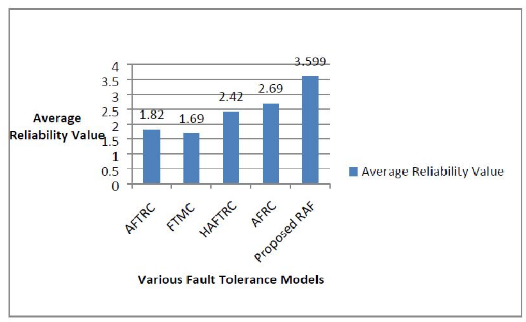

Figure 5: Comparison of Proposed RAF with Existing Models w.r.t their Average Reliability Value

In the above figure 5, average reliability value of proposed framework is the highest amongst the existing fault tolerance models.

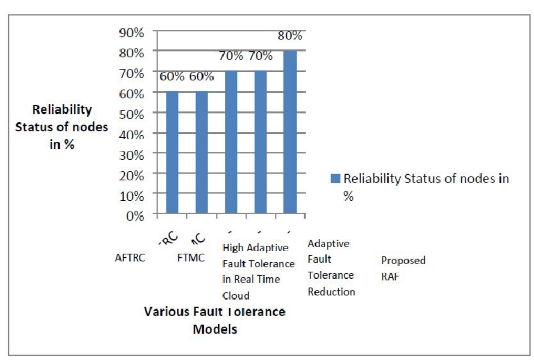

Figure 6: Comparison of Proposed RAF with Existing Models w.r.t their Average Reliability Status of Nodes

The graph 6 describes the reliability status of virtual nodes and in case of proposed reliability framework maximum number of virtual nodes is passed in comparison with existing models of fault tolerance.

7. CONTRIBUTION

The contribution of the work presented in the proposed research is to develop the fault tolerance framework to enhance the reliability of virtual nodes by reducing the number of failure nodes. The reliability of virtual nodes is evaluated using the reliability assessment algorithm, decision mechanism algorithm with the support of backward recovery using fine-grained checkpointing and replication algorithm. Reliability of virtual nodes is evaluated with respect to cost-effectiveness, throughput and self recovery. The processing speed of virtual nodes is also increased. Failure rate of virtual nodes is decreased and also failure nodes are revived in next computing cycle using fine-grained checkpointing. The developed framework gives better results than the existing fault tolerance models in terms of availability and response time.

8. CONCLUSION AND FUTURE SCOPE

The proposed reliability framework calculates the reliability of virtual nodes using adaptive fault tolerant method. Backward recovery technique is also implemented to make the system fault tolerant and the calculated reliability value of virtual nodes help in replacing the failure nodes making the cloud platform more reliable and also enhances the performance of cloud nodes.

The work presented in the reliability framework overcomes the limitations of various extensions of existing fault tolerance models in terms of reliability, cost-effectiveness, processing speed and throughput.

Load balancing technique can also be used to balance load of virtual machines so that resources can be shared equally among the virtual machines and failure of virtual machines can be reduced much.

REFERENCES

1. Z. Amin, N. Sethi and H. Singh. **Review on Fault Tolerance Techniques in Cloud Computing,** *International Journal of Computer Applications,* vol. 116, no.18, pp. 11-17, 2015. https://doi.org/10.5120/20435-2768

2. A. Bhardwaj and S. Goundar. **Efficient Fault Tolerance on Cloud Environments – A Survey,** *International Journal of Computers and Applications* vol. 7, no. 3, pp. 20-24, 2018.

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

4. M. Dhingra and N. Gupta (1). **Architectural Framework for Cloud Reliability Model using Fault Tolerance Techniques,** *International Journal of Innovative Technology and Exploring Engineering* vol. 8, issue 11, pp. 515-519, 2019.

5. M. Dhingra and N. Gupta (2). **Algorithm to enhance the reliability of virtual nodes using Adaptive Fault Tolerance Technique,** *The IIOAB Journal,* vol. 10, no.2, pp. 89-94, 2019.

6. M. Dhingra and N. Gupta (3). **Reliability Framework for Cloud Virtual Nodes using Fault Tolerance Techniques,** *International Journal of Engineering and Advanced Technology,* vol. 9, no. 1, pp. 7586-7593, 2019. https://doi.org/10.35940/ijeat.B2502.109119

7. M. Dhingra and N. Gupta (4). **Comparative analysis of fault tolerance models and their challenges in cloud computing,** *International Journal of Engineering and Technology*, vol. 6, no. 2, pp. 36–40, 2017.

8. R. Jhawar and V. Piuri. **Fault Tolerance Management in IaaS Clouds,** *IEEE First AESS European Conference on Satellite Telecommunications (ESTEL),* pp. 1-6, October 2012.

9. R. Jhawar, V. Piuri and M. Santambrogio. **A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing,** *IEEE International Systems Conference (SysCon),* pp. 1-6, March 2012.

10. R. Kaur and M. Mahajan. **Fault Tolerance in Cloud Computing,** *International journal of Science Technology & Management,* vol. 1, no.3, pp. 201-206, 2014.

11. P. Kumari and P. Kaur. **A survey of fault tolerance in cloud computing,** *Journal of King Saud University-Computer and Information Sciences,* pp. 1-18, September 2018. https://doi.org/10.1016/j.jksuci.2018.09.021

12. M. Mao and M. Humphrey. **A performance study on the VM startup time in the cloud,** *In: Proceedinngs of the 2012 IEEE Fifth International Conference on Cloud Computing, CLOUD '12, Honolulu, HI, USA,* pp. 423–430, June 2012.

13. A. Meshram, A.S. Sambare and S. D. Zade. **Fault Tolerance Model for Reliable Cloud Computing,** *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 1, no. 7, pp. 600-603, 2013.

14. H. Mohammad, A. Mohammad, Y. Eman and E. Hanem. **A Study on Fault Tolerance Mechanisms**

in Cloud Computing, *International Journal of Computer Electrical Engineering*, vol. 10, no. 1, pp. 62-71, 2018.

15. P. Padmakumari and A. Umamakeswari. **Methodical Review on Various Fault Tolerant and Monitoring Mechanisms to improve Reliability on Cloud Environmen**t, *Indian Journal of Science and Technology*, vol. 8, no. 35, pp. 626-631, 2015.

16. P. Palaniammal and R. Santosh. **Failure Prediction for Scalable Checkpoints in Scientic Workflows using Replication and Resubmission Task in Cloud Computing,** *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 2, no. 4, pp. 985-991, 2013.

17. H. Reza, Faragardi, R. Shojaee and A. Rajabi. **An analytical model to evaluate reliability of cloud computing systems in the presence of QoS requirements,** *IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS), At Japan*, pp. 315-321, June 2013. https://doi.org/10.1109/ICIS.2013.6607860

18. M. Sheheryar and H. Fabrice. **Adaptive Fault Tolerance in Real Time Cloud Computing,** *IEEE World Congress on Services*, pp. 280-287, November 2011,.

19. P. Srivaramangai and R. Srinivasan. **A Model to provide a Reliable Infrastructure for Cloud Computing,** *Proceedings of the World Congress on Engineering*, pp. 978-988, July 2012, 2(1).

20. T. Zaidi. **Modeling for Fault Tolerance in Cloud Computing Environment,** *Journal of Computer Sciences and Applications*, vol. 4, no. 1, pp. 9-13, 2016.

21. W. Zhao, P. M. Smith and L. Moser. **Low-Latency Fault-Tolerance System,** *Electrical Engineering & Computer Science Faculty Publications*, vol. 56, no. 6, pp. 716-740, 2010.

22. B. Pandey, D. K. Yadav and S S. Soam. **Application Portability Pertaining TOSCA in Cloud Computing Environment,** *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 5, pp. 2252-2259, September-October 2019. https://doi.org/10.30534/ijatcse/2019/61852019

23. K. Suresh , P. P. Reddy and P. Pushkal. **Smart Home Services Using Cloud and Internet of Things,** *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 4, pp. 1560-1567, July-August 2019. https://doi.org/10.30534/ijatcse/2019/79842019

24. G. Kiruthiga and S. M. Vennila. **An Enriched Chaotic Quantum Whale Optimization Algorithm Based Job scheduling in Cloud Computing Environment,** *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 4, pp. 1753-1760, July-August 2019. https://doi.org/10.30534/ijatcse/2019/105842019