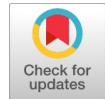


Architectural Framework for Cloud Reliability Model using Fault Tolerance Techniques



Mridula Dhingra, Neha Gupta

Abstract: Cloud Computing is an essential podium for workable and impracticable users. It achieves high quality reliable services provided to the users via data storage servers. The key challenge in cloud architecture is to run the facilities without any hazel to the users. In today's world of information technology, most of the applications are real time. 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:

- Failure to complete the task in prescribed time threshold value.
- Failure to achieve prescribed reliability value.

Virtualization and Internet-based Cloud computing causes diverse types of failures to occur and so necessity for reliability and availability have turn out to be of vital concern. To ensure reliability and availability of cloud technologies, methods for fault tolerance need to be developed and deployed. The proposed work will focus on adaptive behavior during the assortment of replication and fine-grained check pointing methods for achieving a reliable cloud platform that can grip diverse client requests. In addition to this, the proposed work will also conclude the best suitable fault tolerance scheme to each chosen virtual machine.

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

I. INTRODUCTION

Cloud computing is a promising area that provides computing and storage space to the end-users as a service. Cloud computing is well-known for its contribution in numerous areas like research, business, health, e-commerce etc. As the demand of cloud computing is rising day by day, the cloud vendors should provide various services to users to meet up their desired eminence needs. The advantages of cloud architecture are reduction in cost, services on demand, guaranteed quality of service etc. The major issue is to deal with reliability against services to the users and the clients in cloud architecture. Reliability can be defined as a particular job is completed within a specific point of 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, serviceability. Throughput is the amount of data passing through a system or process in a specific time. Completion time of a particular process is known as response time.

Manuscript published on 30 September 2019.

*Correspondence Author(s)

Mridula Dhingra, Research Scholar, Faculty of computer applications, Manav Rachna Institute of Research and Studies, Faridabad, India

Dr. Neha Gupta, Associate Professor, Faculty of Computer, Applications, Manav Rachna Institute of Research and Studies, Faridabad, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

To enhance the reliability, throughput ought to be high and response time must be less. Availability refers to the time required 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 domain, if a method is not able to accomplish the task as it is intended, then the whole structure turn out to be less reliable or there can be the incidence of breakdown. As the computation in cloud is implemented on remote systems, so the possibility of failures become high and the fault tolerance abilities are essential to overcome the system's faults to execute the task correctly as the failure arise.

A. Fault Tolerance Methodology

Like every technology, cloud computing undergoes from some severe issues like fault tolerance of real time systems. The main factors which can cause the failure in real time environment are implementation, recurrent modifications, up gradations, maintenance and exhaustive workload. The reliability of systems will be decreased if the realistic actions are not considered while handling the probable failures in cloud systems. Real time systems are scrutinized on the basis of timeliness and fault tolerance factors. Timeliness is the capability of a real time system to complete the proposed assignment within a specific time and fault tolerance is the ability of the system to work elegantly in the occurrence of fault. Now a days as the requirement of cloud computing is growing day by day, real time systems can be carried out on cloud environment. In most of the real time cloud applications, execution is done on remote cloud computing nodes. So the probability of errors become high because of undecided latency of computing node but on the contrary most of real time systems are safe and extremely reliable and to conquer the reliability of real time systems in cloud computing, there is huge demand of fault tolerance amongst users. To attain reliability the cloud computing, vendors implements a variety of fault tolerance methods to run the system in case of faulty conditions.

B. Need of Fault Tolerance Methodology

- Fault tolerance provides the method or module to do the task effortlessly even though in the incidence of malfunction.
- Fault tolerance assures quality of service for users by definite recital.
- User can rearrange and append software instructions in accordance with their needs.
- 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 collapse.



C. Existing Fault tolerance Methods

The fault tolerance is broadly classified into three categories. These are: Proactive Fault Tolerance Method:-In this method, fault is predicted in advance and it proactively swaps the suspected apparatus and senses the troubles before it actually occurs. Different methodologies of proactive fault tolerance are: software rejuvenation, self-healing, preemptive migration etc. Reactive Fault Tolerance Method:-In this method faults are handled when the failure occurs. Various methodologies are used in reactive fault tolerance are task resubmission, s-guard, job migration etc.

Adaptive Fault Tolerance Method:- One of the fault tolerance methods is adaptive fault tolerance method in which system is monitored at each and every state and reconfigured in order to attain the reliability. In Adaptive Fault Tolerance Method, all events are completed repeatedly, according to the condition and it guarantees the reliability of significant elements. Methodologies which are used in adaptive fault tolerance method are checkpointing, replication, fragmentation etc.

Table I: Comparative analysis between different Fault Tolerance Methodologies (Dhingra & Gupta 2017)

| Methodology | Method | Fault Detected | Application Type |
|-----------------------|--------------------|---|---------------------------------|
| Software Rejuvenation | Proactive | Host, Network Failure | Fault tolerance |
| Self-Healing | Proactive | Node Failure | Load Balancing/ Fault Tolerance |
| Preemptive Migration | Proactive | Process/Node Failure | Load Balancing/ Fault Tolerance |
| Task Resubmission | Reactive | Application Failure /Node Failure | Load Balancing/ Fault Tolerance |
| S-Guard | Reactive | Application Failure /Node Failure | Load Balancing/ Fault Tolerance |
| Job Migration | Reactive | Application Failure/ Process Failure/Node Failure | Load Balancing/ Fault Tolerance |
| Checkpointing | Proactive/Adaptive | Application Failure /Node Failure/ Host Network Failure | Fault tolerance |
| Replication | Reactive/Adaptive | Application Failure /Node Failure | Load Balancing/ Fault Tolerance |
| Fragmentation | Adaptive | Application Failure | Fault Tolerance |

II. RELATED WORK

In cloud computing a lot of work has already been carried out by researchers in real time systems, but still there is a lot of scope to work on it. Sometimes it is difficult to identify the latency of nodes as users do have not the exact idea about the location of processing nodes and because of the dynamic nature of virtual machines, any node can be added or removed at any point. Most of the real time systems needs fault tolerance mechanism. Malik & Rehman (2005) have proposed a model "Time stamped fault tolerance of distributed RTS". In this model, the concept of time stamping with output of real time applications has been introduced but the reliability of virtual nodes using fault tolerance has not discussed so far. A model "Adaptive Fault Tolerance in Real Time Cloud Computing" has been proposed by (Sheheryar & Fabrice 2011). In AFT RC, if any of the node gets failed and does not provide the correct results, the complete model is not producing results.

"Fault Tolerance Model for Cloud Computing" has been discussed by (Meshram, Sambare & Zade 2013) in this model a virtual machine is selected for execution on the basis of reliability and that node can be detached if it that

node is not able to accomplish the task for real time applications. But more parameters should have been introduced in this model for fault tolerance and better results could be produced. Kong, Huang & Lin (2009) has discussed a model for virtual environment but this model is not appropriate to tolerate the fault of real time applications in cloud environment. Damodar & Poojitha (2017) have proposed an adaptive fault reduction scheme which enhances the efficiency of cloud in terms of availability and throughput because of huge number of checkpoints and it offers the reliable environment to cloud computing. A checkpointing scheme is introduced by (Goiri, Julia, Guitart & Torres 2010), in which accumulating time of checkpointing is reduced, distributed file system of Hadoop has been used to save checkpoints. Cao, Simonin, Cooperman & Morin (2015) has proposed checkpointing for fault tolerance, this model supports to the lengthy tasks and tasks with higher priority. Ganga & Karthik (2013) have discussed replication method to tolerate the faults where scientific work flow has used. Das & Khilar (2013) have also discussed the replication technique to amplify the availability of the system, it lessens the imminent faults by adding new nodes.

Table II. Existing Fault Tolerance Models with their procedure and limitations (Dhingra & Gupta 2017)

| Model Name | Procedure for Fault Tolerance | Limitations |
|---|--|---|
| Fault Tolerance Management (FTM) Jhawar & Piuri (2012) | This model depends upon virtualization method to examine fault tolerance that clearly enhances the reliability, availability of systems employed in cloud architecture, Jhawar & Piuri (2012) | In this approach, detailed analysis of fault tolerance services is not mentioned. |
| Low Latency Fault Tolerance (LLFT) Zhao, Smith & Moser (2010) | In this approach, distributed applications for fault tolerance are discussed which have set up inside a cloud architecture. It creates the replica of nodes by using a variety of replication procedures to protect the applications from various faults. Zhao, Smith & Moser (2010) | This approach is expensive in terms of cost for maintaining replicas of virtual machines. |
| Fault Tolerance Model for Cloud (FTMC) Meshram, Sambare & Zade (2013) | It measures the reliability of computing nodes and the node with the highest reliability is selected Meshram, Sambare & Zade (2013) | Additional variables should have been used for result assessment to improve the reliability. |
| Dynamic Adaptive Fault Tolerance (DAFT) Sun, Chang, Gao, Jin & Wang (2012) | In this approach check pointing and replication techniques are implemented to enhance the services of cloud nodes, Sun, Chang, Gao, Jin & Wang (2012) | Backward recovery is not discussed much in this model. |
| Adaptive fault tolerance model in real time cloud computing (AFTRC) Sheheryar & Fabrice (2011). | Backward recovery is focused through check pointing to increase the reliability of nodes. Sheheryar & Fabrice (2011). | In this model, the nodes does not complete the job within threshold time are removed, however they provide the accurate output. |

III. PROPOSED RELIABILITY FRAMEWORK

The proposed model will consist of N virtual nodes. Each node will take the input data from the input buffer and then the input will be given to the each virtual machine. Every time the input will be taken by the node, it will perform the operation and provide the result. All the results will be passed to further modules for reliability calculation and decision making. The proposed reliability framework will primarily divided into three modules, these are:

1. Acceptance Computation
2. Reliability Assessment
3. Decision Mechanis

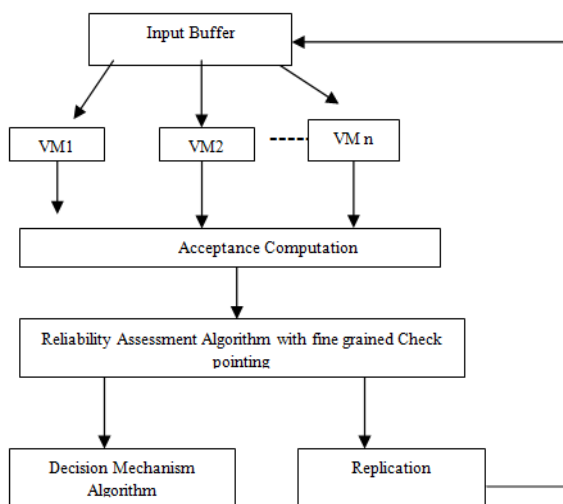


Figure 1. Reliability Assessment Architecture

Acceptance Computation (AC): Each node will receive the input from input buffer, executes and produces the output. The output will be given to AC module. AC module will be 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 will further calculate and verify the threshold time required to complete the task assigned to virtual machine. If the results are found incorrect then backward recovery will be performed through replication technique. If the results will exceeds the minimum threshold time then the node will be considered faulty and backward recovery will be performed through fine-grained checkpointing, otherwise the results will passed to reliability assessment module to measure the reliability of virtual machines and further to decision mechanism module. **Reliability Assessment (RA):** The reliability is a continuous measure and changes its value after every computation cycle. By default, initial value of reliability of each virtual node is 100 %. In this module reliability assessment algorithm (proposed) will be 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 will be forwarded to decision making module. However, if the reliability value of virtual machine will be less than the minimum reliability then the backward recovery will be performed by the system through fine-grained check pointing.

Decision Mechanism (DM): This module will choose the concluding output for a computing cycle. DM module will opt the output of the node with maximum reliability amongst all the computing nodes producing the correct result within threshold time. In this module, if two nodes will produce the identical maximum reliability then the output of the node with smaller IP address will be chosen as the best reliability output.

Architectural Framework for Cloud Reliability Model using Fault Tolerance Techniques

In DM module, system reliability level (SRL) will be predefined by the system. SRL will be the minimum reliability level to be accomplished to pass the result. In this module decision mechanism algorithm (proposed) will be 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. Flowchart to display the working of proposed model is as follows:

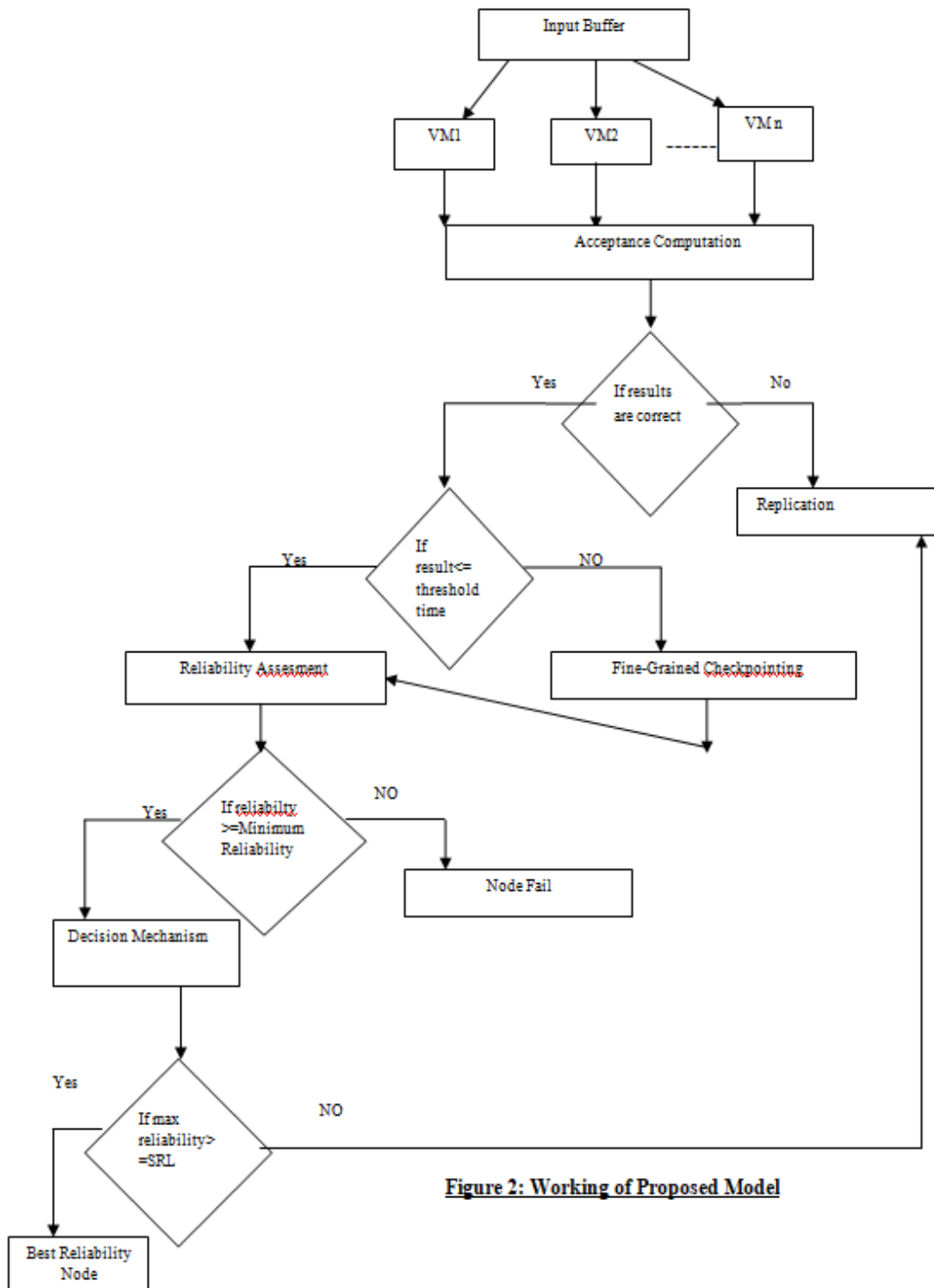


Figure 2: Working of Proposed Model

IV. IMPLEMENTATION

In the proposed framework Cloudsim will be used to perform the results. CloudSim offers a generalised simulation framework that allows faultless model and replication of application execution. CloudSim is a simulation tool that permits developers to check the

performance of their guidelines in a convenient situation and it is free of cost. It is a toolkit for cloud environments which evaluates resource provisioning algorithms.

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. In the proposed framework to measure the results through CloudSim, netbeans will be used and script will be written in Java programming language. This provision will deploy acceptance computation, reliability assessment, decision mechanism. This package will also apply fine-grained checkpointing and replication mechanism (Damodhar & Poojitha 2017). This simulator offers the capability to calculate availability, throughput, time overhead and monetary cost overhead.

V. CONCLUSION

As we know that failures are inevitable in the cloud environment. To solve this problem, in this paper an adaptive fault tolerance method is proposed to provide reliability of cloud nodes. Backward recovery technique will also be implemented to make the system fault tolerant and the calculated reliability value of virtual nodes will help in replacing the failure nodes making the cloud platform more reliable and will also enhance the performance of cloud nodes. The adaptive nature of the proposed method specifies that the cloud's performance will be appreciably enhanced.

REFERENCES

1. M. Dhingra and N. Gupta, "Comparative analysis of fault tolerance models and their challenges in cloud computing", in International Journal of Engineering and Technology, vol 6(2), 2017, pp.36–40.
2. M. Damodhar and S. Poojitha, "An Adaptive Fault Reduction Scheme to Provide Reliable Cloud Computing Environment", in IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19(4), 2017, pp.64-73.
3. I. Gori, F. Julia, J. Guitart and Torres, "Checkpoint-based Fault – tolerant Infrastructure for Virtualized service Provider", in IEEE/IFIP Network Operations and Management Symposium, April 2010, pp. 455-462.
4. A. Meshram, A.S. Sambare and S.D. Zade, "Fault Tolerance Model for Reliable Cloud Computing", in International Journal on Recent and Innovation Trends in Computing and Communication, vol. 1(7), 2013, pp.600-603.
5. A.K. Rai, P. Kumar and P. Manisekaran, "High Adaptive Fault Tolerance in Real Time Cloud Computing", in IOSR Journal of Engineering (IOSRJEN), 2013, vol. 4(3), pp. 24-27.
6. M. Sheheryar and H. Fabrice, "Adaptive Fault Tolerance in Real Time Cloud Computing", in IEEE World Congress on Services, November 10, 2011, pp. 280-287.
7. X. Kong, J. Huang, and C. Lin C., "Comprehensive Analysis of Performance, Fault-tolerance and Scalability in Grid Resource Management System", in Eighth International Conference on Grid and Cooperative Computing, Lanzhou, China, 27-29 August, 2009.
8. R. Jhawar and V. Piuri, "Fault Tolerance Management in IaaS Clouds", in IEEE First AESS European Conference on Satellite Telecommunications (ESTEL), 2 October 2012, pp.
9. R. Kaur and M. Mahajan, "Fault Tolerance in Cloud Computing", in International journal of Science Technology & Management, vol. 1(3), 2014, pp. 201-206.
10. W. Zhao, P. M. Melliar-Smith and L.E. Moser, "Fault tolerance middleware for cloud computing", in Cloud Computing 40 International Journal of Engineering & Technology (CLOUD), 2010 IEEE 3rd International Conference, July 2010, pp. 6774.
11. S. Malik, M.J. Rehman, "Time Stamped Fault Tolerance in Distributed Real Time Systems", in IEEE International Multitopic Conference, Karachi, Pakistan, 2005.
12. J. Cao, M. Simonin, G. Cooperman and C. Morin, "Checkpointing as a service in heterogeneous cloud environments", in Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., Shenzhen, China, May 2015, pp. 61–70.
13. K. Ganga and S. Karthik, "A fault tolerant approach in scientific workflow systems based on cloud computing", in Proc. Int. Conf.

Pattern Recognit., Informat. Mobile Eng. (PRIME), February 2013, pp. 378–390.

14. P. Das, and P.M. Khilar, , "VFT: A virtualization and fault tolerance approach for cloud computing", in Proc. IEEE Conf. Inf. Commun. Tech- nol. (ICT), April 2013, pp. 473–478.
15. D.W. Sun, G.R. Chang, S. Gao, L.Z. Jin and X.W. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments", in Journal of computer science and technology, vol. 27(2), 2012, pp. 256-272.

AUTHORS PROFILE



Mridula Dhingra is a Research Scholar at Faculty of Computer Applications at Manav Rachna International Institute of Research and Studies, Faridabad campus. She has completed her Masters in Computer Applications from Maharishi Dayanand University, Rohtak. She has total of 12+ year of experience in teaching and research. She has authored and coauthored 4 research papers in various National and International Journals.



Dr. Neha Gupta has completed her PhD from Manav Rachna International University and has total of 14+ year of experience in teaching and research. She is a Life Member of ACM CSTA, Tech Republic and Professional Member of IEEE. She has authored and coauthored 34 research papers in SCI/SCOPUS/Peer Reviewed Journals (Scopus indexed) and IEEE/IET Conference proceedings in areas of Web Content Mining, Mobile Computing, and Cloud Computing. She has published books with publishers like IGI Global & Pacific Book International and has also authored book chapters with Elsevier, CRC Press and IGI global USA. Her research interests include ICT in Rural Development, Web Content Mining, Cloud Computing, Data Mining and NoSQL Databases. She is a technical programme committee (TPC) member in various conferences across globe. She is an active reviewer for International Journal of Computer and Information Technology and in various IEEE Conferences around the world.