DOI: http://dx.doi.org/10.26483/ijarcs.v13i4.6883

ISSN No. 0976-5697

Volume 13, No. 4, July-August 2022

International Journal of Advanced Research in Computer Science

RVIEW ARTICLE

Available Online at www.ijarcs.info

A REVIEW ON OPTIMIZATION TECHNIQUE FOR FAULT TOLERANCE IN CLOUD COMPUTING

Nilophar & Ms.Ankita Srivastava
Dept of CSE, Integral University,
Lucknow, India

Abstract: It is the outcome of the emergence of on demand service in big scale distributed computing models. Adaptable technology because it delivers software and resources that can be flexibly scaled up and down. These systems have varying degrees of reliability. Tolerance to unexpected hardware or software failures is measured by a system's fault tolerance. As a result of cloud computing's inability to handle a wide range of errors, its reputation has been tarnished. Tolerating Byzantine errors is difficult since they typically go unnoticed at the beginning and can quickly spread to other virtual machines before being discovered. Thus, vital applications such as air traffic control, online baking, and so on are still avoiding the cloud because of these concerns. As well as some existing models, the goal of this research is to better understand and compare fault tolerance strategies utilised for fault tolerance in cloud systems. The purpose of this work is to evaluate a cloud -based fault-optimization strategy that will address the shortcomings of existing algorithms.

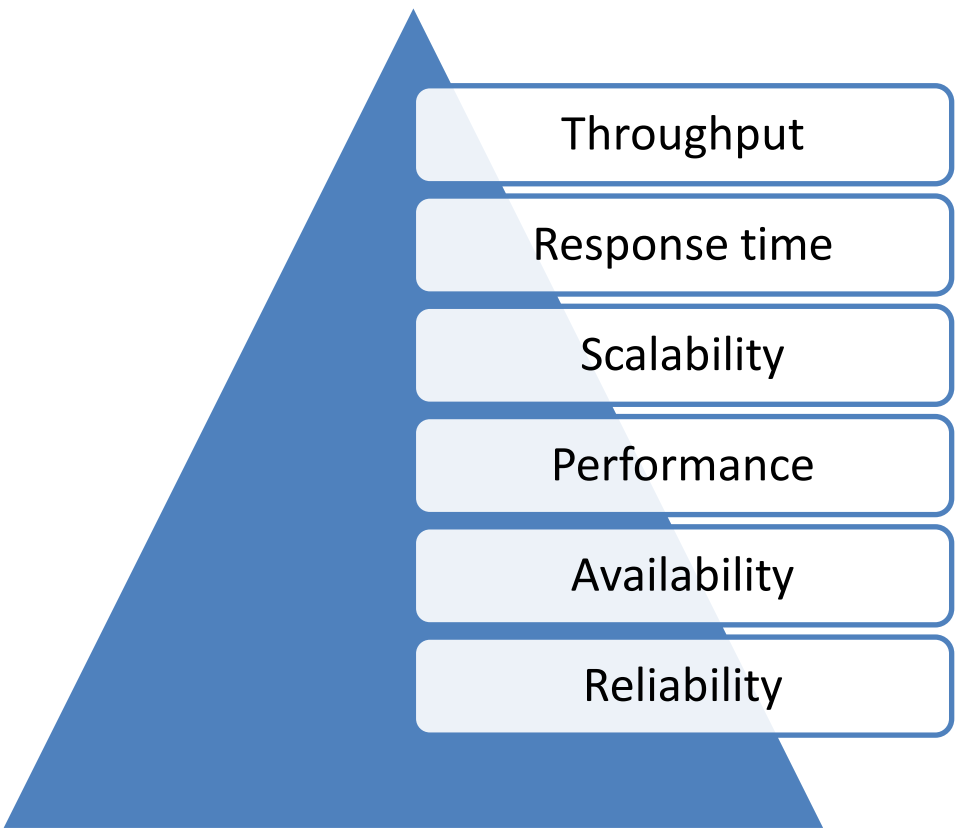
Keywords- Cloud Computing, Fault Tolerance, Dependability

I. INTRODUCTION

Distributed and large-scale computing environments can be found in a cloud computing system. Virtualized and dynamic computing services [1] are available in the pool. Using a subscription model, these services can be accessed through the cloud environment. Cloud services are extremely scalable in nature, dynamically offered to clients, and transparently so that they are not delivered manually. Computing resources can be delivered over the Internet using cloud computing. As an alternative to storing and upgrading software on our own hard drives, we can use a service via the Internet, in a different location, to store our data [2]. The notion of reusing IT resources is at the heart of cloud computing. Virtualization, distributed computing, utility computing, and, more recently, networking, web, and software services, are the pillars around which cloud computing is based. Remotely-managed hardware and software are used by individuals and businesses. Cloud services include file storage, social media, email, and online business applications. In order to use these services, one does not need any knowledge of the underlying hardware and software components. Fault detection, removal, prevention, forecasting, and tolerance are all examples of cloud -based fault handling mechanisms. The critical and initial fault handling mechanism is one of those fault detection mechanisms [4]. However, detecting Byzantine flaws in their early stage is difficult due to their evasive character. A delay-sensitive byzantine fault checking technique was presented in the earlier study, and it was found to be successful in identifying byzantine faults [3]. As the name suggests, another system is in place to deal with the defects at various points in time, either proactive or reactive. Fault prevention is a critical component since it employs defensive mechanisms to help guard against errors [4]. However, fault tolerance is essential in cloud systems as a contingency fault handling method since defects are typically produced to circumvent the present defensive system. Fault tolerance is an effort to maintain service continuity despite the incidence of faults, among other things. Fault tolerance is a realistic approach to improving cloud dependability. Some solutions for fault tolerance include check pointing, replication, task migration, self-healing, safety-bag checks, and retry, task resubmission, reconfiguration, and masking.

II. CLOUD COMPUTING AND FAULT TOLERANCE

It is becoming more and more popular because of the benefits it provides to customers, such as the flexibility with which applications can be deployed, the ease with which resources can be added and freed up to be used for running applications, and the fact that the cost of resources is calculated based on the migration of resources in each use [5]. User's can use virtualized resources through the Internet environment in an Infrastructure as a Service (IaaS) or a Platform as A Service (PaaS) environment, which includes APIs, to develop applications on a specific technology platform, rather than having to purchase and maintain a physical server infrastructure. Additionally, users can take use of Software as a Service (SaaS), which is mostly supplied as a web-based programme that can be accessed from anywhere. In terms of user experience, a computer system achieves dependability when it accomplishes certain tasks while also producing dependable outcomes.When developing cloud computing infrastructure, one of the most important factors to consider is the availability of services. As a result, in the majority of circumstances, it is always required to exercise prudence [6].A cloud computing service must be able to identify and behave appropriately to ensure the smoothness and the quality of service (QoS). Fault is an abnormal condition in one or more parts of the system. A fault may or may not be the cause of the system failure to comply with the original design.In order to properly establish the system's fault tolerance, problem detection, diagnosis, and preventive components are required. Metering, monitoring of variables, and the condition of the system are used for fault detection and diagnosis [7, 13]. Human operators must do data analysis and evaluation on the collected data. Within a knowledge-based approach, the method of fault detection and fault diagnosis might be explored. Methods for detecting and classifying changes in system attributes are frequently employed in comparisons between the measured values in the system and the standard procedure. Signaling models or process models are used to describe the changes in the observed signals. Qualitative information from operators may also be used to construct heuristic symptoms. The task of diagnosing problems is to identify the shape, size, location, and time required to notice them [8, 12]. Classification algorithms can be used to identify defects based on symptom patterns. Inference approaches, on the other hand, can uncover additional details about fault-symptom trees or if-then rules. Experience allows for the calculation of statistical data (such as the mean time to failure and the error probability). Defintion of cloud -level defective tolerance is depicted in figure 1.



,

Figure 1: Metrics for Fault tolerance

III. LITERATURE REVIEW

As previously said, cloud computing has resulted in a shift in the distribution model of information technology from one of a product to one of a service, resulting in the transformation of the information technology industry. In response to this evolution, the internet has enabled the on-demand availability of a wide range of software, platforms and infrastructure resources as scalable services to be made available to users worldwide. When it comes to cloud computing services, their performance is constrained by their inherent susceptibility to errors, which occurs as a result of the huge scale at which they function. It is only via good management of the performance-related challenges of dependability, availability, and throughput by cloud service providers that it is feasible to employ cloud computing services to their fullest potential.Fault tolerance is crucial for optimal performance in cloud computing [9, 10]. Cloud computing allows users to access resources on-demand. Cloud computing allows businesses and users to access software without installing them on physical devices. It also provides Internet access to important resources. A high performance network with high connectivity, interactivity and durability allows for the transfer of information technology from a product to a service model A Study on Fault Tolerance was given by **Dr. J Meena Kumari et. al** Currently cloud computing is becoming increasingly popular, according to the solution. When an application is running on a server, node failure is the primary worry. This is referred to as fault tolerance. Cloud service providers' capacity to deliver high-quality services may be jeopardised if their applications cannot be relied upon or are not available when they need to be. For example, check pointing, Job Migration, Task Resubmission and Replication are all examples of cloud-based fault-tolerance approaches. Fault tolerance is most commonly achieved by replication. In this study, we discuss a replication fault tolerance approach **(Gokhroo et al., 2017).**A system's fault tolerance (FT) is its capacity to continue to execute its intended function despite the presence of defects. When it comes to operational effectiveness and avoidance of breakdowns, FT is key. A FT-based system should be able to evaluate defects in individual software or hardware components, power failures, or other types of unforeseen adversity and still meet its stated requirements **(Dubrova, 2008).Nguyenet al (2013)**developed a decision-making paradigm that is fault tolerant. Two factors are used to measure the reliability of a node: accuracy and time. The system does backward recovery if any of the nodes fails to meet the level. Processing nodes in this model are added or deleted based on how well they react to changes in the environment. A critical issue in distributed computing is addressed in "A fault tolerance method to topology control in distributed sensor networks," by **L. B. Bhajantri and N. Nalini.** To prevent repeated system failures on numerous failure locations, a distributed system employs a separate fault tolerance method. Mechanisms such as replication, scalability, and reliability are discussed in this study. For fault tolerance in mobile agent systems, **Rajwinder Singh and Mayank Dave** suggested a method based on the checkpoint methodology. Faultless and improved performance can be achieved by analysing the graphs that before them. There was also talk of merging graph and non-graph-based techniques to produce a system with great fault tolerance in the future.**A. Munir, J. Antoon,et. al**. if defect is not recognized and recovered in time, it results in a system failure in a real-time distributed system. The primary focus of this research is on real-time distributed system failure detection and recovery strategies. There are several forms of defects, including network faults, physical faults, media faults, processor faults, process faults, and service expiration problems. In real-time distributed systems, an appropriate fault detector and fault tolerance mechanism are employed.To help with proactive fault tolerance in scientific workflow applications, **Anju Bala et al**. proposed the development of intelligent task failure detection models. The model's operation is broken down into two distinct parts. Machine learning techniques are used in the first module to forecast task failures, and in the second module, the actual failures are discovered. Basically two type ofused to make system fault tolerancei.e. Proactive and Reactive fault tolerance

- **Reactive Fault Tolerance**

Reactive fault-tolerance rules lower the work required to recover from failures when the breakdown happens in the real world. A system's resilience is improved by the use of this technology.

- ✓ **Check pointing/ Restart**

Here, the current calculation state is saved. Periodically, normal execution is checked. In case of failure, the operation is rolled back to the last stable storage state. This strategy allows the system to recover and resume regular functioning. There are two ways to rollback. Using checkpoints or logs for rollback recovery. The former just utilises check pointing, whereas the latter also logs non-deterministic occurrences.

- ✓ **Replication**

Replication is a typical fault tolerance strategy in cloud environments. It improves resource availability in dispersed storage environments. Because the storage system is so big and complicated, replica selection and placement are major issues. So allocating an identical number of copies in heterogeneous would not increase speed but will waste extra storage space. Data may be accessed even when part of the copies are unavailable due to replication.

- ✓ **Job Migration**

It has happened on occasion that a project could not be entirely completed on a specific machine owing to a variety of factors. It is possible to move tasks from one computer to another in the event of failure of any job. Job migration may be accomplished with the use of a HA-Proxy.

- ✓ **S Guard**

It is It's based on rollback. An approach for commodity server clusters was proposed by several writers. Stream processing is less likely to be disrupted due to checkpointing. SGaurd also leverages modern file distribution systems like GFS and HDFS. Although using checkpoints is costly, SGaurd increases performance by utilising modern distributed file systems.

- ✓ **Retry**

This job level strategy is the most straightforward of the bunch. On the same cloud resource, the user re-submits his or her assignment.

- ✓ **Task Resubmission**

When a failed task is recognized in a job, the job may fail. In this scenario, the task is resubmitted to the same or a different resource for execution at the moment of failure.

- ✓ **User Defined exception Handling**

In this section, the user specifies the particular action to be taken in the event of a task failure in workflows.

- **Proactive Fault Tolerance**

Proactive fault tolerance rules hinder recovery from errors. Some of these policies are proactive. Replace suspect nodes with new ones, preventing issues. It predicts failures based on the system log and hardware or software state.

- ✓ **Software Rejuvenation**

Software rejuvenation strategies are required to prevent or eliminate software ageing. The software rejuvenation strategies rely on restarting the service to restore it to a youthful condition. The software rejuvenation approaches only treat the symptoms of software ageing.

✓ **Proactive Fault Tolerance using Self Healing**

Systems can be rebooted, restarted or re-diagnosed using the same approach to diagnose and prescribe alternative remedies. The healing outcome will be sent to the diagnosis modules for learning. To achieve peer healing, the technique not only applies various prescription for diverse types of probable consequences, but leverages other hosts.

✓ **Proactive Fault Tolerance using Preemptive Migration**

A continuous observation and analysis of the application is carried out in this approach. The feed-back-loop control technique is required for preemptive migration of a job.

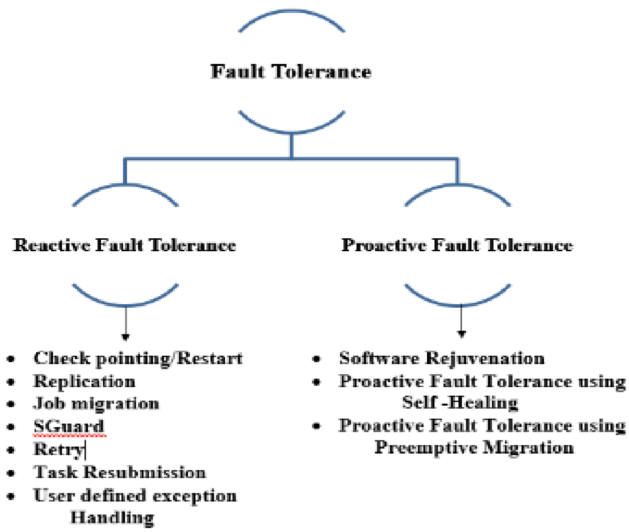


Figure 2: Fault tolerance Approaches [9]

IV. SCOPE OF STUDY

According to the research gaps identified, there is a possible need for developing autonomic fault tolerance in cloud environments by utilizing a variety of various characteristics. These are the numerous difficulties that academicians have encountered in implementing fault tolerance into cloud computing, as identified in the literature review:

- Cloud heterogeneity makes it difficult to pinpoint the source of a problem. There is a pressing need to develop more effective methods for identifying and correcting errors.
- Because processing takes place on faraway computers, there are additional opportunities for mistake.
- User organizations have no control over data centre failures, hence an autonomous fault tolerance solution must be implemented for applications running in a cloud environment.
- Because cloud environments are dynamically expandable, unpredictable, and frequently virtualized resources are supplied as a service, it is challenging to grasp the changing system status.
- The high level of system complexity, users are only given with limited information, making it difficult to build an appropriate fault tolerance solution for them.
- Fault Prediction and Monitoring framework needs to be developed for real time applications that execute in cloud environment.

V. CONCLUSION

The dynamic nature of the cloud environment makes it susceptible to errors and system failures due to unanticipated system behaviour. Cloud computing resilience and dependability may be improved by assessing and responding to errors quickly. It is one of the most difficult aspects of making a system fail-safe. cloud computing has become a reliable calculating stage for both business and non-business computation consumers due to the enormous rise of the internet and its users, as well as its wonderful possibilities in simplicity, quality of service, and on-interest administrations. Adoption is possible due to the fact that it delivers software and resources that can be dynamically scaled. As a result of the constantly changing nature of the cloud , different unanticipated errors and malfunctions occur. Fault tolerance refers to a system's capacity to gracefully respond to an unanticipated equipment or programming breakdown. Failings must first be identified, appraised, and dealt with appropriately if cloud computing is to be reliable and resilient. It has been suggested that various defect detection methods and architectural models can be used to improve cloud 's fault tolerance.

VI. REFERENCES

[1] Geeta, &Prakash, S. (2018). A review on quality of service in cloud computing. Advances in Intelligent Systems and Computing, 654, 739–748. https://doi.org/10.1007/978-981-10-

[2] Amin, Z., Singh, H., &Sethi, N. (2015). Review on Fault Tolerance Techniques in Cloud Computing. International Journal of Computer Applications, 116(18), 11–17. https://doi.org/10.5120/20435-2768

[3] Qasem, G. M., &Madhu, B. K. (2018). A Classification Approach for Proactive Fault Tolerance in Cloud Data Centers. International Journal of Applied Engineering Research, 13(22), 15762–15765. http://www.ripublication.com

[4] Kumari, P., &Kaur, P. (2021). A survey of fault tolerance in cloud computing. Journal of King Saud University - Computerand Information Sciences, 33(10), 1159–1176.

[5] Dhingra, M., & Gupta, N. (2019). Reliability framework for cloud virtual nodes using fault tolerance techniques. International Journal of Engineering and Advanced Technology, 9(1), 7586–7593.

[6] Amin, Z., Singh, H., &Sethi, N. (2015). Review on Fault Tolerance Techniques in Cloud Computing. International Journal of Computer Applications, 116(18), 11–17.

[7] Meshram, A. D., Sambare, A. S., &Zade, S. D.(2013). Fault Tolerance Model for Reliable Cloud Computing.

[8] Nguyen, H., Shen, Z., Tan, Y., &Gu, X. (2013, July).FChain: Toward black-box online fault localization forcloud systems. In Distributed Computing Systems

(ICDCS), 2013 IEEE 33rd International Conferenceon (pp. 21-30). IEEE.

[9] https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.7415&rep=rep1&type=pdf

[10] R. K. and A. Nanoti, "A Review on Genetic Algorithm for Fault Tolerance in Distributed Sensor Networks," Ijarcce, vol. 5, no. 12, pp. 346–348, 2016, doi: 10.17148/ijarcce.2016.51281.

[11] L. B. Bhajantri and N. Nalini, "A fault tolerance approach to topology control in distributed sensor networks," Proc. 2012 IEEE Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2012, no. 978, pp. 208–212, 2012, doi: 10.1109/ICACCCT.2012.6320772.

[12] Malik, S., &Huet, F. (2011, July). Adaptive FaultTolerance in Real Time Cloud Computing. In Services(SERVICES), 2011 IEEE World Congress on (pp. 280-287). IEEE.

[13] M. Trivedi, A survey on resource provisioning using machine learning in cloud computingInt. J. Eng. Dev. Res., 4 (4) (2016), pp. 546-549Gokhroo et al., 2017

[14] M.K. Gokhroo, M.C. Govil, E.S. Pilli Detecting and mitigating faults in cloud computing environment**,** IEEE Int. Conf. (2017)

[15] A Fault Tolerant Scheduling System Based on Checkpointing for Computational Grids Mohammed Amoon Computer Science and Eng. Dept., Faculty of Elec. Eng., Menofia University, Egypt King Saud University, P. O. Box: 28095- 11437 Riyadh-Saudi Arabia m_amoon74@yahoo.com Singh and Cabillic, 2003, Singh, P., Cabillic, G., 2003. A Checkpointing Algorithm for Mobile Computing Environment, 65–74.

[16] Singh et al., 2016, S. Singh, Y.S. Jeong, J.H. Park **A** survey on cloud computing security: issues, threats, and solutions J. Netw. Comput. Appl., 75 (2016), pp. 200-222

[17] A. Munir, J. Antoon, and A. Gordon-Ross, "Modeling and analysis of fault detection and fault tolerance in wireless sensor networks," ACM Trans. Embed. Comput. Syst., vol. 14, no. 1, 2015, doi: 10.1145/2680538.

[18] Anuj, Zeeshan, Harshpreet Singh, and Nisha Sethi. "Review on fault tolerance techniques in cloud computing." International Journal of Computer Applications 116.18 (2015).

