INTERNATIONAL JOURNAL OF
RESEARCH IN COMPUTER
APPLICATIONS AND ROBOTICS
ISSN 2320-7345

# A COMPARATIVE STUDY OF FAULT TOLERANCE TECHNIQUES IN CLOUD COMPUTING

**Seyedeh Solmaz Madani[1], Shahram Jamali[2]**

*[1] Department of computer engineering, Ardebil Branch, Islamic Azad University, Ardebil, Iran*
*[2] Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran*
*Author Correspondence: Iran, Solmaz_sm66@yahoo.com*

**Abstract: -** Cloud computing is a method of computing in a place that provides users with the capabilities of information technology as a service and allows them to have access to these services on the Internet without having specialized information or controlling the infrastructure. Due to the exponential growth of cloud computing, the need for fault tolerance in the cloud is a key factor for investigation. Fault tolerance has all the necessary techniques to keep active power and reliability. The main advantages of using fault tolerance in cloud computing include failure recovery, lower costs, and improved performance criteria. The motivation to examine existing techniques and models of fault tolerance in cloud computing has encouraged researchers to participate in the development of more efficient algorithms. So, in recent years, there has been a lot of research on fault tolerant systems. This has led to new strategies to find out the benefits and barriers of fault tolerant systems in cloud computing. This paper introduces various fault-tolerance methods recently used in cloud computing and a comprehensive review is presented of various techniques and models in the field of error tolerance in cloud computing. By studying and analyzing fault-tolerant techniques, according to the requirements and criteria we can use the perfect fit for error tolerance in cloud computing.

**Keywords***:* cloud computing, fault tolerance, reliability, failure.

## 1. Introduction

Cloud computing is a concept that refers to applications and services which are run on a distributed network with the help of available resources. Cloud computing when software and applications are run provides an abstract representation of physical systems [1]. The main advantage of cloud computing is to provide reliability, low cost, high availability, scalability and flexibility for end users which appears as a new computing paradigm [2]. Service quality also plays an important role in cloud computing in which the efficiency and reliability of the service can be considered as two important aspects. Efficiency in cloud computing means that if a user requests a particular service, how fast can it is answered by the user's request? And reliability in cloud computing has meant this whether the requested user service has been successfully handled by him or not. Each of performance and reliability play an important role in cloud computing, because if the service reliability is low, it occasion frequent crashes in the cloud service, which in turn results in a reduction in the number of customers and in the result is a loss for the server. If the reliability of the service is not high, but its efficiency is low, users who ask for services should wait for a long time and this will also be a disappointment to them. So it can be directly related to tolerance to error [3] [1]. Tolerance to fault is used In order to improve reliability in cloud computing. Fault tolerance is one of the important issues in cloud computing and is related to all the necessary techniques to enable the system to tolerate the remaining software fault in the system after its development. Fault-tolerance

techniques provide reliability and validity in the cloud environment. The main advantages of the implementation of the fault tolerance technique in cloud computing are: failure recovery, low cost, improved performance criteria, and so on [4]. In this paper, we study the techniques of error tolerance in cloud computing and the key stimulation for this study is to identify different models and methods of fault tolerance in cloud computing which will support researchers in creating a more efficient algorithm So this article is organized into the following sections;
After presenting the introduction in Section 2, we introduce the error tolerance in cloud computing. Section 3 describes the types of faults. In Section 4, the types of fault tolerance are expressed in cloud computing. Part 5: fault-tolerance requirements are explained and in Section 6, are investigated the reasons for the accurate analysis of various fault-tolerant techniques and models. Part 7 relates to fault tolerance criteria, and Part 8 and 9, respectively, examine techniques and models of fault in cloud computing. Finally we gave the conclusions and references.

## 2. Fault tolerance

Fault tolerance is a feature of the system that prevents a computer system or network device from failing due to any fault or failures in system execution. The fault tolerance includes effective steps to prevent such errors or failures in the system [5]. In fact, a fault-tolerant system is capable of providing the service in question in an efficient manner if one or more faults or failures occur in system components and the availability and reliability will not be lost of the system. A tolerable bug system is a form that can In the event of a bug, tolerate it and continue to work. Perhaps it's better to have a definition of error at first, because with this word, two words are also mentioned in the mind that is the fault and the failure. But there are three differences between them. Failure: A failure occurs when an expected system is not functioning correctly so, if the system misconduct affords the system to fail at least one of its capabilities properly, then the system is in a malfunction.

**Fault:** The cause of the failure is a fault in the system. So the fault is a physical malfunction or a failure of a hardware or software component.

**Bug:** The afford of an error is a bug in the system
"Not necessarily all bugs will not lead to faults"
Failure, faults and bugs may occur in applications, virtual machines, and even hardware. The system must be capable of handling the fault and continue to operate. Figure 1 shows the fault path.

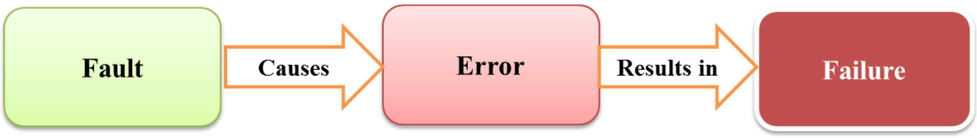

**Figure (1):** The path to failure [1]

So there are two solutions to this problem:
**Fault detection:** To provide each evaluation, the first step that a system must perform is identifying the fault functions.

**Fault Repair:** After the system detects a fault, the next step is to avoid the fault or to improve it.

## 3. Fault types

The cloud platform has three layers: hardware, virtual machines, and applications. Each of them has a malfunction. These failures can be present on any hardware or virtual machine layer during program execution. Therefore, due to the nature of the failure, should be carry out appropriate action .faults can be Split up in cloud computing in several categories [7] [6].
**Network faults:** faults regarding to networks are known as network faults. This fault occurs when the data is not destined for various reasons such as packet loss, closed date, destination failure, link failure, and so on
**Physical fault:** faults that occur in hardware, such as CPU crashes, memory crashes, crash failures, and so on.
**Media fault:** Mistakes that arise due to the lack of communication media.
**Processor fault:** A fault occurred on the processor due to operating system failure.
**Process fault:** faults afford by lack of resources or software bugs.

**Service termination fault:** The service life of the resource is over, but the application still requires the use of resources.
**Transient:** This kind of fault lasts for a long time and only once appears and after the actions are done it was disappear. For example, at first the network message from the origin to the destination cannot reach its destination, but after a while it reaches its destination prosperity.
**Alternate:** Alternate faults are those that repeatedly and alternately distributed. These failures are not good, mainly due to the failure of each component or function inappropriately between the components, for example a faulty connection.
**Stable:** This kind of failure still exists in the system after defective systems are repaired or replaced in some cases completely.

## 4. Types of fault tolerance

Fault tolerance can be classified in two categories of hardware fault tolerance and software fault tolerance. Figure 2 shows this category:

### *4.1 Hardware fault tolerance*

One of the main goals of fault tolerance is make the computer system which can automatically recover if multiple random faults occur in hardware components. The developed methods for this work generally include the partitioning of a computational system in several modules. Each module in the system has been redundant Therefore, if the failure occurs in one of the modules, the backup module will continue to work. Fault tolerant methods include two types of error handling and dynamic recovery [4].

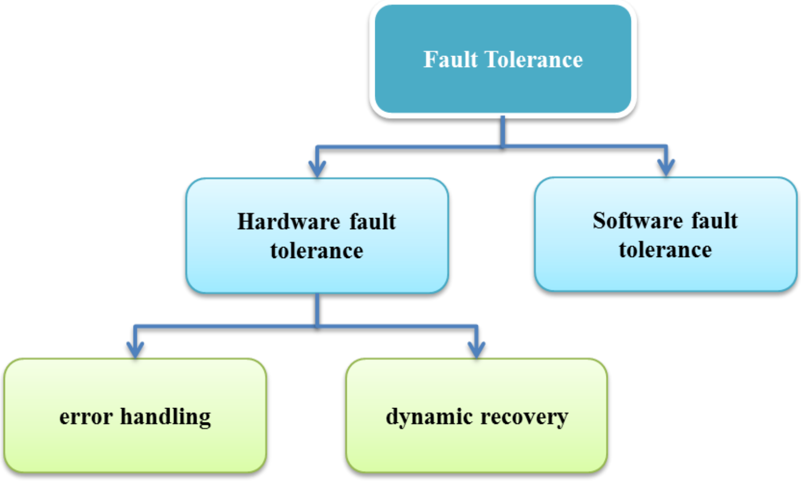

**Figure (2):** Types of Fault Tolerance

**Fault coating:** fault coating is a structural redundancy technique that completely eliminates faults in a set of mixed components. A number of identical components execute similar functions and their output is voted on to remove faults afforded by a defective module.
**Dynamic retrieval:** A dynamic retrieval technique is only used when a copy of the work or calculations is made to run at a time. This technique administer self-repair. Like a fault-coating technique, additional spare components are used to perform backup operations (preventive redundancy).

### *4.2 Software fault tolerance*

Software faults (programming faults) can be exploited using static and dynamic methods similar to those used for hardware fault handling. One of these methods is n-version programming, which uses static redundancy in the form of independent programs. All of them are doing the same thing. There is another method called design variation, which incorporated software and hardware fault tolerance by applying a fault-tolerant computer system using hardware and software in redundant channels [8].the main target of the design diversity technique is to tolerate hardware and software faults, but the cost is very expensive.

## 5. Fault-tolerant requirements

An important goal in designing distributed systems is to create a system in such a way that it can automatically repair minor defects without generally affecting the overall system performance. In particular, whenever a malfunction happened, the system must be repaired in an acceptable action. In other words, a distributed system is expected to tolerate the fault. To realize the role of tolerable fault, there should be covered a number of useful requirements for distributed systems including the following [9].

**Average Failover Time (MTTF):** The waiting time for failure bearing in mind that the system has been utilizable.

**Average Repair Time (MTTR):** This mentioned to the expected time to repair the system after a failure.

**Average Breakdown Time (MTBF):** This represents the average time for the next failure and is calculated as follows:

$$MTBF=MTTF+MTTR \qquad (1)$$

**Reliability:** Points to the point that a system can run continuously and without fail.

$$Reliability = \frac{MTTF}{1+MTTF} \qquad (2)$$

**Availability:** The system operates at any one time and is available to perform tasks.

$$Availability = \frac{MTBF}{1+MTBF} \qquad (3)$$

**Safety:** refers to a situation when nothing works temporarily, nothing happens.

**Maintenance:** refers to how to repair a failed system. A superb maintenance system may also have a high degree of availability.

$$Maintainability = \frac{1}{1+MTTR} \qquad (4)$$

## 6. Reasons for accurate analysis of various techniques and models of fault tolerance in cloud computing

Implementation of fault tolerance in cloud computing due to its complexity, reliability, and the following reasons are confronting challenges which requires detailed analysis.

A. There is a need to implement a self-governed fault tolerance technique for several instances of an application running on multiple virtual machines.

B) Different technologies from cloud computing providers and vendors will require a formidable integrated system

C) With new approaches in the cloud, should also be expand fault tolerance techniques and scheduling algorithms

D) Self-governed fault tolerance techniques must be aligned with different clouds.

E) To ensure maximum reliability and availability must be operational several providers of these services [10].

## 7. Criteria for fault tolerance in cloud computing

Fault tolerance methods in cloud computing considers various parameters [11]:

**Adaptive:** The entire process is accomplished automatically according to the work conditions in real time.

**Efficiency:** used to check the system efficiency. Efficiency should be at an acceptable cost, such as reducing the response time to users as well as improving latency in response.

**Response time:** The amount of time spent responding by a particular algorithm. This parameter must be minimized.

**Scalability:** The ability of an algorithm to perform error tolerance for a system with a finite number of nodes. This criterion needs to be improved.

**Operating Capacity:** To count the work that has been completed. This benchmark must be high to improve system performance.

**Reliability:** The purpose of this aspect is to give a true or acceptable result in a limited environment.

**Availability:** The probability that an item will be satisfactory at a specified point within the time required under the declared conditions. The availability of a system is typically measured as a reliability factor, as it increases the reliability of access.
**Usability:** The percentage of a product that can be exploited by a user to achieve goals with effectiveness, efficiency, and satisfaction.
**Related overhead:** the tolerance of the fault distinguished the amount of overhead captive in the execution of an algorithm. This overhead is performed due to overhead to work, processor inside, and communication between the processes. This overhead should be reduced so that an error-tolerant technique can work effectively.
**Effect on cost:** The cost here is defined only as a warning cost.

## 8. Fault Tolerance Policies in Cloud Computing

Based on fault tolerance, different techniques and strategies are classified as follows:

### *8.1 Tolerable Preventive fault*

The policy of tolerance is the action-oriented fault, preventing faults and failure by their prediction and the actionist replaces the dubious component. It detects the problem before it happens. Some techniques based on this policy contain preventive migration, software rejuvenation, self-healing, and so on.

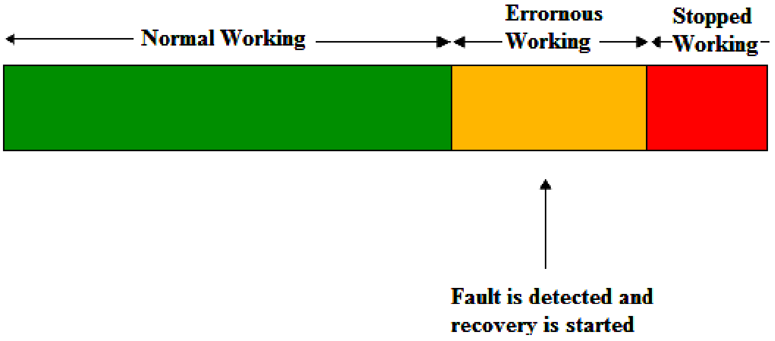

**Figure (3):** Timetable for a precautionary fault detection system [7]

**Software rejuvenation:** Software rejuvenation is a technique that designs the system for periodic reload. This technique returns the system to its original state and helps a new start.
**Preventive Migration:** Preventive Migration counts on the feedback loop control mechanism. The application is continuously monitored and analyzed.
**Self-repair:** A great job can be divided into several parts. This method is used to improve performance. When different instances of an application are running virtually on different devices, this feature automatically manages application sample defects.

### *8.2 Tolerable reaction fault*

Responsive fault tolerance policy, when failure happens effectively attempt to reduce failures. This method creates more power for the system. A number of different techniques are based on this policy; Inspection / restart, re-assignment of function, release of workflow, handling of user's specific behavior, retrial, labor migration, S-Guard, etc. [14] [13] [12].

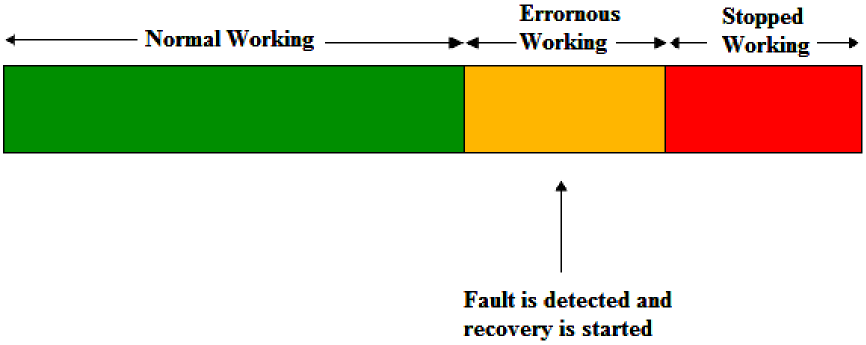

**Figure (4):** Timeline for a reactive fault detection system [7]

**Inspection:** An efficient task fault handling Technique for long and large tasks. In this method, is made a careful examination after any change in the system.
**Work migration:** Sometimes it happens that for some reason, a work cannot be done on a particular machine. At the time of the failure of any work, one can transfer the work to another machine.
**Duplication:** Duplication means copying. Various tasks are repeated and implemented in order to succeed and get the favorable result on different resources.
**Examining safety packages:** In this case, blocking of commands that does not have safety properties.
**S-Guard:** fewer disturbances to normal processing flow. It is based on work retrieval and rollback.
**Retry:** In this case, we will re-implement and retrieve a task. This is the easiest technique to re-work again in one source.
**Task Resubmission:** Whenever a task fails, it is determined. In this case, at runtime, the task is sent again to run to the same source.

### *8.3 Adaptive fault tolerance*

The tolerance of an application fault requires the change depending on the range of control inputs and current position in its space. Adaptive tolerance automatically adjusts the instructions to the status control. And ensuring the credibility of critical modules under any resources and time constraints Provides as much redirection resources and modules as possible.

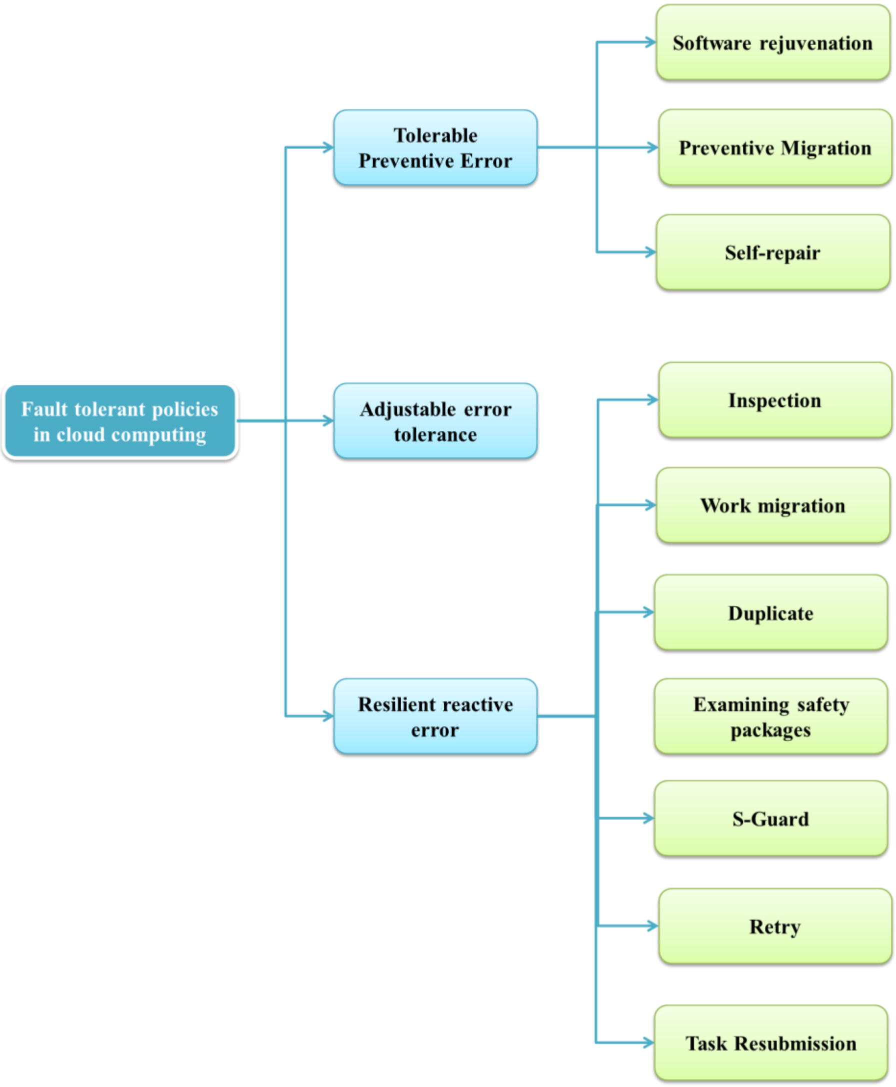

**Figure (5):** Classification of Fault Tolerance Policies in Cloud Computing

**Table (1):** Comparison of different tolerance techniques

| Politics | System | Programming framework | Environment | Fault detection | Application type | Techniques |
|---|---|---|---|---|---|---|
| Preventive / Reactionary | HAProxy | Java | Virtual machine | Node failure or processing | Fault tolerance and load balancing | **Self-healing, job migration, replication** |
| Preventive | SHelp | SQL,JAVA | Virtual machine | Application crash | Fault tolerance | **Check Pointing** |
| Preventive / Reactionary | Assure | JAVA | Virtual machine | Network / host failure | Fault tolerance | **Check Pointing, retry, self-healing** |
| Preventive / Reactionary | Hadoop | Java,HTML, CSS | Cloudy environment | Node downtime and application های programs کاربردی | Volume of data | **Job Migration, Replication, S-Guard** |
| Preventive / Reactionary | Amazon EC2 | Amazon Machine Image, Amazon Map | Cloudy environment | Node downtime and application programs | Fault tolerance and load balancing | **Replication, S-Guard, Task Resubmission** |

## 9. Fault tolerant models in cloud computing

Nowadays, various models are presented based on the techniques mentioned in the previous section. Most of them are based on these techniques.

**FT-Cloud:** Cloud computing applications are usually provided on a large scale and with complexity. However, unfortunately, their reliability is still far from ideal [15].In reference [16] it is a chart-based ranking component which uses its architecture to build cloud applications. It is a two-stage framework of operational algorithms that includes ranking and fault tolerance. FT-Cloud provides cloud computing applications to deal with faults. This fault-tolerant model offers cloud computing against collision faults and amounts.

**BFT-Cloud:** Repeat policy is used in this model. This model can be categorized as a reactive model. When a request is located in the cloud computing system, the request must be performed in different nodes. One of the nodes is selected as the main node and the other nodes are selected as backup nodes. In the application, all applications run on the device locally. If the results on the backup nodes are the same, the output is correct and correctly requests the module. If one of the backup copies or major editions failed, they will give a different response than other outputs. In this case, this node is known as a defective node and in the upgrade phase, should be done the recovery operation. If the node is a defective node, it will change it to the original new version at the initial update stage and if the defective node is one of the backup nodes, then it should be replaced with convenient nodes at the replacement update stage [17].

**FTM model:** Another architecture that deals with cloud computing fault tolerance is FTM, which is used to manage fault tolerance [18]. The FTM model is one of the reaction techniques that use three techniques (job migration, checkpoint / retrieval, repeat). This model guaranteed reliability and flexibility by using creative methods that the user can identify and enforce to achieve a satisfactory level of fault without having to know about its accomplishment.

**LLFT model:** The model introduced in [19], which is called LLFT, delivers fault tolerance with a low postponement. This model has provided F.T the capacity to develop distributed programs or data centers. LLFT uses the leader / follower duplication approach. In this structure, copies of a process form the group. In each group, a process is selected as the main process, and the rest are selected as the backup. These groups are a process that provides services for users in the form of a server group. Communication between groups happened through a virtual connection. The original version of the source group sends messages to the original version in the destination group through multicast via virtual communication. The original version of the goal group provides for the implementation of the administer orders to perform uncertain operations performed by backup copies.

**Candy model:** In reference [20], has been introduced a component-based distributed modeling framework which is a useful and comprehensive semi-automatic model and is described by the language of the system model. This model indicates that cloud services and cloud computing providers should guaranteed availability, which is one of the highlights of cloud services.

**AFTRC Model:** A fault tolerance model for cloud computing is based on a real-time system can benefit from computing capacity and cloud-scalable virtual environment for better real-time use. In this model, the fault system is tolerant of action .Based on the reliability of the processor nodes; it exports an executable command [21].

**FTWS model:** The FTWS model stated in [22], fault tolerance is presented using startup and propagation techniques according to the priority of tasks. This model plans the workflow with a deadline in the presence of faults. This model is also based on the fact that the workflow is a set of processed tasks with a segregated order based on data and affiliation.

**Magi-Cube Model:** In [23], the Magi-Cube architecture is introduced which a very reliable storage structure for cloud computing. They use this system on top of HDFS and handle it as a storage system for reading / writing files and managing cloud data. They also create a startup file and modify the component to work independently in the background. In fact this model is based on the three contradictory components of the storage system that they are reliability and high efficiency and low cost (space).

**Table (2):** Comparison of policies applied in each of the models

| Tolerable models for cloud computing | Policies and techniques | | | | |
|---|---|---|---|---|---|
| | Preventive | | Reaction | | |
| | Self-repair | Preventive Migration | Inspection | Duplicate | Labor migration |
| **BFT-Cloud** | × | × | × | ✓ | × |
| **FTM** | × | × | ✓ | ✓ | ✓ |
| **LLFT** | × | × | × | ✓ | × |
| **FTWS** | × | × | ✓ | ✓ | × |
| **Candy** | × | × | × | ✓ | × |
| **Magi Cube** | × | × | × | ✓ | ✓ |
| **AFTRC** | × | × | ✓ | ✓ | ✓ |

**Table (3):** Comparison between different models based on error tolerance criteria in cloud computing

| BFT-Cloud | Magi-Cube | FT-CLOUD | CANDY | FTM | FTWS | LLFT | AFTRC | Model<br>Criterion |
|---|---|---|---|---|---|---|---|---|
| Yes | Yes | Yes | Yes | No | No | No | Yes | **Adaptive** |
| High | High | High | Average | Average | Average | High | High | **Performance** |
| Average | Average | Average | Average | Average | Average | Average | Average | **Response time** |
| High | High | High | High | Low | Low | High | High | **Scalability** |
| High | High | Average | High | Average | Low | Average | High | **Throughput** |
| High | High | High | High | Average | Average | High | High | **Reliability** |
| High | Average | Average | High | High | Average | High | High | **Availability** |
| Average | High | High | Average | Average | Average | Average | High | **Usability** |
| Average | Average | High | Low | Low | High | Low | Average | **overhead** |
| High | High | High | Low | Low | High | Low | Average | **Impact on cost** |

## 10. Conclusion

Cloud computing has become a commonly used computing technology, which very popular. There must be reliability and availability for users. This requires utilize of tested tolerance methods which can manage any kind of fault in every aspect. The fault tolerance is required when a fault enters the system borderline, Therefore, fault tolerance techniques are used to predict these failures and take the necessary actions before the damage happened. Reliability and availability are two important parameters in cloud computing. Therefore, we need a fault tolerance method that will prepare the services provided in cloud computing against the resulting faults and failures.in this paper, we discussed the need for fault tolerance, covering various techniques for implementing fault tolerance. There are a number of fault-tolerant techniques in the cloud, which prepare various fault tolerance mechanisms by increasing the reliability of the system. Also, these techniques represent a major role in providing service availability to the user. But there are still some issues and deficiencies that must be considered for each framework. There are debility that none of them can complete all aspects of the faults. Therefore, it is possible to dominate the debility of all previous models and try to create a proper and efficient model that covers the most aspects of fault tolerance. In the future, it is also expected to better understand the types of faults in hardware, software, and cloud infrastructure by providing other models of architecture with higher fault tolerance, higher reliability, availability, and more impressive performance.

## REFERENCES

[1] Himanshu Agarwal, Anju Sharma, "A Comprehensive Survey of Fault Tolerance Techniques in Cloud Computing", International Conference on Computing and Network Communications (CoCoNet), IEEE, 2016.

[2] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in Proceedings of the1st ACM symposium on Cloud computing, ACM, pp. 193 –204, New York, USA, 2010,.

[3] Bo Yang, F. Tan and Y.S. Dai, "Performance Evaluation of Cloud Service Considering Fault Recovery," The Journal of Supercomputing, vol. 65, no. 1, pp. 426-444, 2013.

[4] V.M.Sivagami, K.S.EaswaraKumar, "Survey on Fault Tolerance Techniques in Cloud Computing Environment", International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-9, December 2015.

[5] Pankaj Deep Kaur, Kanu Priya, ".Fault Tolerance Techniques and Architectures in Cloud Computing -A Comparative Analysis", International Conference on Green Computing and Internet of Things (ICGCIoT), 2015.

[6] Abhishek Bhavsar, Ameya More, "A Holistic Approach to Autonomic Self-Healing Distributed Computing System" International Journal of Computer Applications (0975 – 8887) Volume 76– No.3, August 2013.

[7] HARPREET KAUR, AMRITPAL KAUR, "A SURVEY ON FAULT TOLERANCE TECHNIQUES IN CLOUD COMPUTING", International Journal of Science, Engineering and Technology, Volume 3 Issue 2, 2015.

[8] Amritpal Singh, Supriya Kinger, "An Efficient Fault Tolerance Mechanism Based on Moving Averages Algorithm" © 2013, IJARCSSE, ISSN: 2277 128X .

[9] Salma M. A. Ataallah, Salwa M. Nassar, "Fault Tolerance in Cloud Computing – Survey", 11th International Computer Engineering Conference (ICENCO), IEEE, 2016.

[10] Bala.A, Chana.I.(2012),"Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing" International Journal of Computer Science Issues.

[11] Seyyed Mansur Hosseini and Mostafa Ghobaei Arani, "Fault-Tolerance Techniques in Cloud Storage: A Survey", International Journal of Database Theory and Application, Vol.8, No.4 (2015), pp.183-190.

[12] Anju Bala, Inderveer Chana, "Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing", IJCSI international Journal of Computer Science Issues, Vol.9, Issue 1, No 1, January 2012 .

[13] Patel et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(12), December - 2013, pp. 573-576.

[14] Wenbing Zhao, P.M. Melliar and L.E. Mose" Fault Tolerance Middleware for Cloud computing" 2010 IEEE 3rd International Conference on Cloud Computing.

[15] Nazari Cheraghlou Mehdi, Khadem-Zadeh Ahmad, Haghparast Majid , "A Survey of Fault Tolerance Architecture in Cloud Computing", *Journal of Network and Computer Applications,* Volume 61, Pages 81–92, February 2016.

[16] Zheng, Z., Zhou, T.C., Lyu, M. R., and King, I., "FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications " 2010 IEEE 21st International Symposium on Software Reliability Engineering. DOI: 10.1109/ISSRE.2010.28, 1-4 Nov. 2010, p. 398 - 407.

[17] Zhang, Y., Zheng, Z., Lyu, M. R. "BFTCloud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing" Cloud Computing (CLOUD), 2011 IEEE International Conference on 4-9July (2011), 444 – 451.

[18] Kaur, J., Kinger, S. "Analysis of Different Techniques Used For Fault Tolerance", IJCSIT: International Journal of Computers & Technology, Vol.4 - Issue 2, (2013), 737-741.

[19] Zhao, W., Melliar P.M. and Mose, L.E., " Fault Tolerance Middleware for Cloud Computing" 2010 IEEE 3rd International Conference on Cloud Computing. DOI: 10.1109/CLOUD.2010.26, 5-10 July 2010, p. 67 -74.

[20] Machida, F., Andrade, E., Seong Kim, D. and Trivedi, K.S., "Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML" 2011 30th IEEE International Symposium on Reliable Distributed Systems. DOI: 10.1109/SRDS.2011.33, 4-7 Oct. 2011, p. 209 - 218.

[21] Sudha Lakshmi, S. "Fault Tolerance in Cloud Computing", ACICE, Vol.04, Special Issue 01, 2013.

[22] Jayadivya, S.K., JayaNirmala, S., SairaBhanus, M., "Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing" International Journal on Computer Science and Engineering (IJCSE). Vol. 4 No. 06 June 2012.

[23] Feng, Q., Han, J., Gao, Y., Meng, D. "Magi-cube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing" 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage. 2012, PP. 89 - 93.