http://dx.doi.org/10.21172/ijiet.144.04

Fault Tolerance in Cloud Computing: A Major Research Challenge

Manoj Kumar Malik[1], Ajit Singh[2], AbhishekSwaroop[3]
[1]*Assistant Professor, Department of IT, Maharaja Surajmal Institute of Technology, Delhi, India*
[2]*Professor, Department of CSE, BipinTripathi, Kumaon Institute of Technology, Uttrakhand, India*
[3]*Professor, Department of IT, BhagwanParshuram Institute of Technology, Delhi, India*

Abstract- Cloud computing is a buzz word nowadays among the computer industry, academia and researchers. The scalability and economic efficiency of the cloud make it popular among the user. However, data security, reliability and dependability are the major concerns among the cloud users. If some fault or failure occurs in a cloud computing system, how it affects the overall performance and working of the system is significant for any cloud computing system. Hence, it can be concluded that fault tolerance is a major issue in cloud computing. This paper presents a brief survey of the major techniques employed in a cloud computing environment to achieve fault-tolerant capabilities. The techniques to handle hardware, as well as software fault tolerance, have been covered in the present exposition.
Keywords: Cloud, Failure, Fault Tolerance, Dependability

I. INTRODUCTION:

Distributed computing is asset sharing on a more prominent scale so to speak free and decreased cost which gives the conveyance of processing assets over the web. Rather than overseeing data all alone framework or updating applications for client needs, one can use assistance over the web arrange, at an elective zone, to store the information and use applications introduced at some remote spot [1]. The essential thought of distributed computing is engaged around the re-usability of data innovation capacities. Distributed computing decreases the running time of employment and reaction time; it likewise limits the hazard in application arrangement. Distributed computing brought down the expense of organization with diminishing exertion and expanding development. The a great many server expanded the throughput with least hazard and lower capital speculation for the framework.

The significant worldview required for distributed computing is based upon circulated figuring, utility processing, organizing, virtualization, web and programming administrations. The clients use programming and equipment oversaw by outsiders situated in some remote area [2]. Online record stockpiling, webmail, long-range informal communication locales and online business applications are some regular cloud administrations. The client can utilize these administrations without knowing the essential or key equipment and programming subtleties.

1.1 Cloud Components:

Distributed computing is comprised of different components which have a particular reason and assumes noteworthy jobs. The cloud segments can be delegated Customers, Server farms and Disseminated Servers [3].

Client: The client is commonly the PC gadgets utilized by the end customers. The end client utilized these gadgets to oversee and keep the data on mists (Cushions, Cell Phones, workstations and so on.).

Data centers: The data centres are gathering of servers where the administration is facilitated. The virtualization is utilized to make a number of the virtual server on a solitary physical server in the server farm.

Distributed servers: The disseminated servers will be servers which are accessible at various geological areas. A circulated server gives better security and openness to the end client.

1.2 Cloud Characteristics:

The major characteristics of cloud computing are given below [4].

Client driven interface: The cloud interfaces are free on the area of the client. They can be gotten to by set up interfaces, for example, web programs and web administrations.

Self-sufficient framework: In the independent framework the necessities of clients can be reconfigured. The client can join programming and data as needs be in an independent framework.

Free coupling: The different assets are inexactly bound as a single asset. The usefulness of inexactly limited asset barely influences the working of another asset.

Versatility and on-request benefits: In this administration, the client is given the on-request asset and administration over the web. The assets gave are adaptable over different information servers.

Dependable Conveyance: To convey the data between assets the TCP/IP is utilized. The cloud foundation utilized the Private System Conventions and the clients are associated through the HTTP convention.

Asset pooling: The pools of assets shared by the quantity of clients are given by the cloud specialist organization. In this administration, the different virtual machines having a place with various clients might be facilitated by a physical server, and it alludes as multi-inactivity.
High Security: The high-security trademark is kept up on the above-clarified attributes. The uncovering of itemizing of usage is maintained a strategic distance from by the deliberation and virtualization of cloud supplier. The Free coupling trademark empowers the errands to execute well, regardless of whether a segment of the cloud is devastated.

1.3 Models of Clouds:
The cloud suppliers offer the different Administrations. These administrations can be assembled into the following three classes [2], [5].
Software as a Service (SaaS):The product as administration model of distributed computing gives the total application to the client on request. The execution of a single occasion at the backend adjusted the various end clients. The suppliers of Saas are Microsoft, Google, Salesforce, Zoho and so forth.
Platform as a Service (Paas):
The stage as administration model offers the product advancement condition as administration. In this model, a stage is given to the end clients in whom blend of use servers is utilized.
Infrastructure as a Service (Iaas):The fundamental stockpiling and processing abilities are given as standard administrations to the client in Framework as an assistance model. To deal with the remaining task at hand the numerous clients shared the different assets accessible.

1.4 Types of Clouds:
The clouds can be classified into the following types based on their access [2], [6].
Public Cloud:It alludes as accessibility of registering assets to clients associated with the web on "Pay As You Go Basis".To deliver the super economic services to the customers the third parties operate and own the Public clouds.
Private Cloud: The exclusive computing services available to the particular group of the user in an organization referred to as the private cloud. The limited and specific access to the group of user in an organization is a private cloud. In a private cloud, the concern is always on data security.
Hybrid Cloud:Thehybrid cloud can be referred as a combination of the private and public cloud. The computing adaptability can be expended by the cloud service provider in a hybrid cloud by using the other cloud provider partially of fully.
Community Cloud: The hybrid clouds are formed bysharing the cloud functionalities (public and private) with similar prerequisites in an organization which reduces the capital investment by dividing the cost among the different cloud users. The community clouds operations may be within the premises or outside.

II. FAULT TOLERANCE

The fault in a system is a phenomenon that leads to the deviation of the system from its expected behaviour. The fault may lead to the failure of the system [7]. The failures may be classified as transient, permanent and intermittent depending upon the time for which failure exists in the system. Fault Tolerance can be defined as asystem design methodology that allows a system to keep workingwithout failure even when some fault occurs in the system. Alternatively, the capability of agile reaction against the programming breaks down and equipment's unexpected behaviour can be defined as fault tolerance.In the event that a framework isn't completely operational, the adaptation to non-critical failure ability may enable a framework to keep working with the decreased limit as opposed to totally closing down after a disappointment.

2.1 Types of Faults:
There are several factors on the basis faults can be classified. Based on faulty computing resources, faults can be classified as follows [8].
Network fault:A Fault that happens in a system because of connection disappointment, arrange parcel, Packet Loss, Packet debasement, goal disappointment, and so forth.
Physical faults: This fault comes about in hardware due to fault in CPUs, power failure, memory fault, storage fault, etc.
Media faults:Media fault takes place due to media head crashes.
Processor faults:The processor fault takes place due to operating system crashes, etc.
Process faults: A fault that comes about due to inefficient processing capabilities, low availability of the resource, software errors, etc.

Service expiry fault: The resource's service time may expire during application is using it.
Fault Tolerance Techniques:
There are various fault tolerance techniques which can be used to provide fault tolerance capability to any computing system [9]. The prevalent fault tolerance techniques are as follows:

2.2 Reactive fault tolerance
Reactive fault tolerance also called as on-demand fault tolerance. When a failure occurs in the system during the execution of an application the reactive fault tolerance policies minimize its effect in the system. Following are the various techniques which are based on reactive fault tolerance policy.
Checkpointing/ Restart -In this technique, the recent checkpointed state is used to start a failed task instead of starting it from beginning.
Replication- The different resources are used to run the various task replicas. In this technique, for successful execution and desired result the replicated tasks run on different machines until the complete replicated task is not crashed.
Job Migration-In job migration technique the task may be migrated on a different machine on the occurrence of failure.
S-Guard-S-Guard depends on rollback recuperation which is less wild to typical stream preparing. It makes more assets accessible. It very well may be executed in HADOOP, Amazon EC2.
Retry- This is the simplest technique in which user can submit the failed taskagain on the same cloud resource.
Task Resubmission-On detection of a failed task, it is either resubmitted to the same machine or the different machine. This is a widely used fault tolerance technique in current scientific workflow systems
User-defined exception handling-with this policy the user can specify the action or treatment for the failed taskfor workflows.
RescueworkflowThis method allows the workflow to continue even if the work fails until progress without addressin g the failed task becomes impossible.
Proactive Fault Tolerance
The proactive replacement of suspected components with other healthy working components is the basic principle of proactive fault tolerance policy. The policy predicts the problem before it comes into the system. It avoids recovery from the problem. Based on this policy followings are the few techniques:
Software Rejuvenation- The software rejuvenation technique designs the system for periodic reboots. In this technique, the system restartswith a clean slate.
Self-Healing- The application instances failures handled automatically in self-healing technique while multiple application instances running on multiple VMs.
PreemptiveMigration- A control mechanism is used in preemptive migration technique which is based on a feedback loop. The applications are continuously analyzed and monitor with this mechanism in the preemptive migration.

III. CHALLENGES OF IMPLEMENTING FAULT TOLERANCE TECHNIQUES IN CLOUD COMPUTING:

The cloud is an abstract representation of a huge network of resources. The size of network resources, volume, storage capacity and capability of processing neither be specified nor be limited in cloud system. In the cloud environment, the geographical position of resources is not known to the user. Hence therefore due to the complexity and inter dependability, the cloud computing system needs careful consideration and analysis to provide the fault tolerance. Followings are more reasons to be considered: [10].

- The implementation of Autonomic fault tolerance technique for multiple instances running on various VMs is to be needed.
- The key issue for designing fault-tolerant cloud computing system is interoperability. Hence to establish a reliable system the integration of different technologies from various cloud infrastructure provider is to be required.
- The integration of available workflow scheduling algorithms and fault tolerance techniques is required with the new possible approach.
- In cloud computing environment the performance of fault tolerance components is compared with other similar components. Hence to develop a method for benchmarking is to be needed.
- The dependent software stack should not be used with various cloud computing provider to ensuring high availability and reliability.
- There must be synchronization among different clouds. Autonomic fault tolerance must react accordingly in the absence of synchronization.

To measure the fault tolerance performance in a cloud computing environment the various parameters like scalability, response time, security, reliability, usability, throughput, availability and associated over-head are considered for the available techniques.

IV. FAULT-TOLERANT MODELS IN CLOUD COMPUTING:

Based on available techniques following are the fault-tolerant models which can be implemented [11].

AFTRC:It is referred to as Adaptive Fault Tolerance model Real-time Cloud Computing [12].The virtual machine's or the processing node's reliability is the basic characteristic to decides fault tolerance for the system. The reliability of nodes is adaptive and changes after every computing cycle. The reliability of a virtual machine or the processing nodes increases if nodes produce the desired result within the specified time limit, if not then the reliability decreases in the AFTRC model.

LLFT:Thedeployment of fault tolerance for the distributed applications in the cloud computing scenario is provided by the Low Latency Fault Tolerance model [13]. The robust replica consistency in LLTF model is maintained transparently for those applications that involve multiple interacting processes.On a fault occurrencein system, the LLFT reconfigured with low latency and mechanism of recovery ensure the existence of backup replica for normal message delivery operations.The semi-active of semi-passive replication approach to protecting the applications against the various faults are provided by the middleware.

FTWS:Based on the priority of the tasks the replication and resubmission of tasks are carried out in fault tolerance workflow scheduling algorithm to provide the fault tolerance [14]. The data and the control dependency are the key factors for deadline workflow schedule in FTWS model when a fault exists. The workflow scheduling considering the task deadline and task failure in cloud environment is a challenging process.

Candy:the candy is the components based availability model. It is the major characteristic and critical challenging issue for the cloud service provider that high availability of components is ensured in this model architecture. The systems modelling language (SysML) [15] is used to express the specifications of candy (component-based availability modelling framework) system. Availability model components translated from SysML diagram are assembled and synchronized to form whole availability model according to stereotype allocations.

FT-Cloud:Itis an architecture model to build the fault-tolerant cloud applications based on components ranking [16]. This architecture model includes two parts;first rankingsection to assign the ranks to components based on the calculated significant value of cloud components second optimal fault tolerancesection to select the optimized fault tolerance strategy for each significant component. The component invocation relationship and invocation frequencies are the major criteria to identify the significant components. The response time constraint specified by the designer to select the fault tolerance strategy for a significant component.

Map-reduce: In the map-reduce model the task is converted into the smaller parts. All the smaller parts are located at different nodes and process the work simultaneously [17]. The result of the individual parts combined to achieve the final output. A feedback strategy is used in this architectural structure to migrate the part process to different machine when running process faces any problem to produce the desired result.

BFT-Cloud:In BFT-Cloud (Byzantine fault tolerance architecture) is classified as reactive architecture which used the Replication policy [18]. The same input is distributed among the available nodes out of which one is selected as a primary node and rests as backup nodes. If the result of primary and all backup nodes for the executed application is same then the output for requesting model is correct. The different answer of any of the node in the architectural model is treated as a fault and node as a faulty node. To detect the x faulty node the network should have 3x+1 node as the architecture's fault detection capacity is approximately 33%.

Gossip architecture:The gossip architecture use decision vector which is an advantage over the BFT architecture. It is introduced to enhance the performance of BFT [19]. In gossip architecture, the replication policy exploits the fault detection and capability of fault tolerance in the cloud computing environment. Every node selects a neighbour node hence two nodes to have to update their decision vector at a time during the processing. The fault detection reliability in the gossip architecture increases by up to 50% which was 33% in BFT architecture. X faulty nodes can be identified in this architecture if 2x+1 nodes are there in the system.

MPI (Message Passing Interface): MPI is the architecture model which uses the reactive method [20]. It is a model for parallel programming uses job migration and checkpoint/restart techniques. The two-layered structure of the MPI model in which upper layer infrastructure communication is not dependent and in the lower layer, it is specified that the backup of a checkpoint is required or not. The running job is migrated to the healthy node on receiving a positive response from the faulty node using these techniques in this model.

V. TOOLS UTILIZED FOR IMPLEMENTING FAULT TOLERANCE TECHNIQUES IN CLOUD COMPUTING:

Adaptation to internal failure difficulties and procedures has been executed utilizing different apparatuses. Followings are the significant instruments which are utilized to execute distinctive adaptation to non-critical failure strategies Based on their programming system, condition and application type [9][13].

Amazon Elastic Compute Cloud: Amazon EC2 offers the versatile registering limit and gives the offices to dispatch and oversees API based server examples utilizing accessible devices [21]. It gives the capacity to keep the case in various areas. Amazon gives the virtual figuring condition, preconfigured bundles that empower a client to run Linux based applications. It empowers to assemble shortcoming tolerant frameworks that work with easeand the base measure of human collaboration. Amazon offers tied down administrations to the client on-request premise through the virtual private cloud.

Assure:Guarantee introduces the salvage focuses which are the areas in existing application codes for dealing with developer foresaw disappointment [22]. It tends to be utilized to give an elective pathway which instigates programming and recoup from programming disappointments by utilizing the blunder virtualization methods to drive a mistaken return utilizing a watched an incentive in a capacity. Guarantee utilizes a generation framework and helps dependent on the need for the framework to execute salvage focuses and mistake virtualization.

Hadoop:Hadoop is open-source java-based programming; gives a dependable and versatile system for disseminated figuring. The two segments, HDFS (Hadoop disseminated record framework) runs on a hub of server bunch to get to information code and Map Reduce Engine to perform map-diminish activities by separating the information into littler pieces. The information is recreated and put away at various machines to give adaptation to internal failure by means of replication method [23].

S-Help:S-help is a lightweight programmed framework that can endure programming disappointment in a virtual machine system. S-help functions as a blunder handler in the distributed computing condition. At first, zero as weight worth is allotted to each save point and expanded when a shortcoming recognized in the framework [24]. The moved back to the most recent checkpoint is chosen by this salvage esteem.

HA –Proxy:It is alluded to as a high-accessibility intermediary. It is an open-source programming device which gives a heap adjusting among an enormous pool of web servers by disseminating the heap. It gives the adaptation to non-critical failure ability to the framework. The Ha-Proxy handles the solicitation by diverting them to another server when an issue happens in the framework.

VI. CONCLUSION

Fault Tolerance Techniques	Policies	System	Programming Framework	Environment	Fault Detected	Application Type
Replication, S-Guard, Task Resubmission	Reactive/ Proactive	AmazonEC2	Amazon Machine Image, Amazon Map	Cloud Environment	Application/node failures	Load balancing, fault tolerance
Self-Healing, Job Migration, Replication	Reactive/ Proactive	HA-Proxy	Java	Virtual Machine	Process/node failures	Load balancing Fault Tolerance
Checkpointing	Reactive	S-Help	SQL, JAVA	Virtual Machine	Application Failure	Fault tolerance
Job Migration, Replication, S-Guard, Resc	Reactive/ Proactive	Hadoop	Java, HTML, CSS	Cloud Environment	Application/node failures	Data-intensive
Checkpointing, Retry, Self Healing	Reactive/ Proactive	Assure	JAVA	Virtual Machine	Host, Network Failure	Fault tolerance

Cloud computing is the promising exemplification for giving specialized benefits as processing utilities. Be that as it may, the remote area of the specialist co-op, the client is defenseless to the unwavering quality and constancy of the registering procedure. Consequently, giving adaptation to non-critical failure in a cloud computing framework is a significant research challenge. In this paper, significant research difficulties in giving adaptation to non-critical failure in cloud computing frameworks alongside flaw tolerant strategies have been talked about. The methods to

accomplish programming just as equipment adaptation to internal failure have been introduced. The mainstream devices for actualizing shortcoming tolerant systems in cloud condition have additionally been examined and their examination is condensed in table 1.

Table 1: Tools Used for Fault Tolerance in Cloud Computing

VII. REFERENCES:

[1] Rimal, B. P., Choi, E., & Lumb, I. (2009). A Taxonomy and Survey of Cloud Computing Systems. NCM, 9, 44-51
[2] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 1(1), 7-18.
[3] Amin, Z., Singh, H., & Sethi, N. (2015). Review on fault tolerance techniques in cloud computing. International Journal of Computer Applications, 116(18).
[4] Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010, September). The characteristics of cloud computing. In Parallel Processing Workshops (ICPPW), 2010 39th International Conference on (pp. 275-279). IEEE.
[5] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
[6] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58.
[7] Tchana, A., Broto, L., & Hagimont, D. (2012). Fault tolerant approaches in cloud computing infrastructures. In Proceedings of the 8th International Conference on Autonomic and Autonomous Systems (ICAS'12) (pp. 42-48)
[8] Sivagami, V. M., & Kumar, K. E. (2015). Survey on Fault Tolerance Techniques in Cloud Computing Environment. International Journal of Scientific Engineering and Applied Science (IJSEAS), 1(9), 419-425.
[9] Bala, A., & Chana, I. (2012). Fault tolerance-challenges, techniques and implementation in cloud computing. IJCSI International Journal of Computer Science Issues, 9(1), 1694-0814.
[10] Essa, Y. M. (2016). A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation. International Journal of Computer Applications, 138(13).
[11] Cheraghlou, M. N., Khadem-Zadeh, A., & Haghparast, M. (2016). A survey of fault tolerance architecture in cloud computing. Journal of Network and Computer Applications, 61, 81-92.
[12] Malik, S., & Huet, F. (2011, July). Adaptive fault tolerance in real time cloud computing. In Services (SERVICES), 2011 IEEE World Congress on (pp. 280-287). IEEE.
[13] Patra, P. K., Singh, H., & Singh, G. (2013). Fault tolerance techniques and comparative implementation in cloud computing. International Journal of Computer Applications, 64(14).
[14] Poola, D., Ramamohanarao, K., & Buyya, R. (2014). Fault-tolerant workflow scheduling using spot instances on clouds. Procedia Computer Science, 29, 523-533.
[15] Chandrakala, N., & Sivaprakasam, D. P. (2013). Analysis of Fault Tolerance Approaches in Dynamic Cloud Computing. International Journal of Advanced Research in Computer Science and Software Engineering, 3(2).
[16] Zheng, Z., Zhou, T. C., Lyu, M. R., & King, I. (2010, November). FTCloud: A component ranking framework for fault-tolerant cloud applications. In Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on (pp. 398-407). IEEE.
[17] Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: a prospective study. New Generation Computing, 28(2), 137-146.
[18] Zhang, Y., Zheng, Z., & Lyu, M. R. (2011, July). BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing. In Cloud Computing (CLOUD), 2011 IEEE International Conference on (pp. 444-451). IEEE.
[19] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In Grid Computing Environments Workshop, 2008. GCE'08 (pp. 1-10). Ieee.
[20] Hawilo, H. (2015). Elastic Highly Available Cloud Computing.
[21] Buyya, R., Yeo, C. S., & Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In High-Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on (pp. 5-13). Ieee.
[22] Takabi, H., Joshi, J. B., & Ahn, G. J. (2010). Security and privacy challenges in cloud computing environments. IEEE Security & Privacy, 8(6), 24-31.
[23] Huerta-Canepa, G., & Lee, D. (2010, June). A virtual cloud computing provider for mobile devices. In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (p. 6). ACM.
[24] Ganesh, A., Sandhya, M., & Shankar, S. (2014, February). A study on fault tolerance methods in cloud computing. In Advance Computing Conference (IACC), 2014 IEEE International (pp. 844-849). IEEE.