"Bodh", BPIT's International Journal of Technology & Management
ISSN : 2454-8421, Volume 1 Issue IV, July to December 2017

Fault Tolerance in Cloud Computing: A Major Research Challenge

Manoj Kumar Malik[1] Ajit Singh[2] Abhishek Swaroop[3]

[1]Assistant Professor, Department of IT, Maharaja Surajmal Institute of Technology, Delhi, India

[2]Associate Professor, Department of CSE,BipinTripathi, Kumaon Institute of Technology, Uttrakhand, India

[3]Professor, School of Computing Science and Engineering, Galgotias University, Greater Noida U.P., India

Abstract:
Cloud computing is a buzz word now a days among the computer industry, academia and the 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 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.

Key words: Cloud, Failure, Fault Tolerance, Dependability

1. Introduction:

Cloud computing is resource sharing on a greater scale in amanner of location independent and reduced cost which provides the delivery of computing resources over the web. Instead of managing information on our own system or upgrading applications for user needs, one can utilize a service over the web network, at an alternative area, to store the data and use applications installed at some remote place [1]. The fundamental idea of cloud computing is focused around reusability of information technology abilities. Cloud computing reduces the running time of job and response time;it also minimizes the risk in application deployment. Cloud computing lowered the cost of deployment with decreasing effort and increasing innovation. The thousands of server increased the throughput with minimum risk and lower capital investment for infrastructure.

The major paradigm required for cloud computing are built upon distributed computing, utility computing, networking, virtualization, web and software services. The users use software and hardware managed by third

Parties locatedat some remote location [2]. Online file storage, webmail, social networking sites and online business applications are some common cloud services. The user can use these services without knowing the basic or fundamental hardware and software details.

1.1. Cloud Components:

Cloud computing is made up of various elements which has a specific purpose and plays significant roles. The cloud components can be classified as Clients, Data Centers and Distributed Servers [3].

Clients: The clients are typically the computer devices used by the end users. The end user used these devices to manage and keeping the information on clouds (PADs, Mobile Phones, laptops etc.).

Data Center: The data centers are group of servers where the service is hosted. The virtualization is used to create number of virtual server on a single physical server in data center.

Distributed servers: The distributed servers are servers which are available at different geographical locations. A distributed server provides better security and accessibility to the end user.

1.2. Cloud Characteristics:

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

User-centric interface: The cloud interfaces are independent on location of user. They can be accessed by established interfaces such as internet browsers and web services.

Autonomous system: In autonomous system the requirements of users can be reconfigure. The user can combine software and information accordingly in an autonomous system.

Loose coupling: The various resources are loosely bound as single resource. The functionality of loosely bounded resource hardly affects the functioning of another resource.

Scalability and on-demand services: In this service the user are provided the on-demand resource and service over internet. The resources provided are scalable over various data servers.

Reliable Delivery: To deliver the information between resources the TCP/IP is used. The cloud infrastructure used the Private Network Protocols and the users are connected through the HTTP protocol.

Resource pooling: The pools of resources shared by the number of users are provided by the cloud service provider. In this service the various virtual machines belonging to different users may be hosted by a physical server, and it is referred as multi-latency.

High Security: The high security characteristic is maintained on the above explained characteristics. The exposing of detailing of implementation is avoided by the abstraction and virtualization of cloud provider. The Loose coupling characteristic enables the tasks to execute well, even if a component of cloud is destroyed.

Models of Clouds:

The cloud providers offer the various Services. These services can be grouped into following three categories[2], [5].

Software as a Service (SaaS): The software as service model of cloud computing provide the complete application to the user on demand. The execution of single instance at the backend serviced the multiple end users.The providers of Saas are Microsoft, Google, Salesforce, Zoho etc.

Platform as a Service (Paas): The platform as servicemodel offers the software development environment as service. In this model a platform is provided to the end users in whom combination of application servers is used.

Infrastructure as a Service (Iaas): The basic storage and computing capabilities are provided as standard services to the user in Infrastructure as a service model. To manage the workload the multiple users shared the various resources available.

1.3. Types of Clouds:

The clouds can be classified in to the following types on the basis of their access [2], [6].

Public Cloud: It refers as availability of computing resources to users connected to internet 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 user in an organization referred as private cloud. The limited and specific access to the group of user in an organization is a private cloud. In private cloud the concern is always on data security.

Hybrid Cloud: Thehybrid cloud can be referred as a combination of private and public cloud. The computing adaptability can be expended by the cloud service provider in 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.

2. Fault Tolerance

The fault in a system is a phenomenon that leads to the deviation of the system from its expected behavior. 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 agilely reaction against the programming break down and equipment's unexpected behavior can be defined as fault tolerance. If a system is not fully operational, the fault tolerance capability may allow a system to continue working with the reduced capacity rather that completely shutting down following a failure.

2.1. Types of Faults:

There are several factors on the basis faults can be classified. On the basis of faulty computing resources faults can be classified as follows [8].

Network fault: A Fault that occur in a network due to link failure, network partition, Packet Loss, Packet corruption, destination failure, etc.

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 resource, software errors, etc.

Service expiry fault: The resource's service time may expire during application is using it.

2.2. 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:

2.2.1. Reactive fault tolerance

Reactive fault tolerance also called as on demand fault tolerance. When a failure actually occurs in the system during execution of an application the reactive fault tolerance policies minimize its effectin the system. Following are the various techniques which are based on reactive fault tolerance policy.

Check pointing/ Restart -In this technique the recent check pointed 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 till the complete replicated task is not crashed.

Job Migration-In job migration technique the task may be migrated on different machine on occurrence of failure.

S-Guard-S-Guard is based on rollback recovery which is less tumultuous to normal stream processing.It makes more resources available.Itcan be implemented 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 failed task, it is either resubmitted to the same machine or to the different machine. This is widely used fault tolerance technique in current scientific workflow systems

User defined exception handling-with this policy the user can specifies the action or treatment for the failed task for workflows.

Rescue workflow-This technique allows the work flow to continue even if the task fails untilit becomes impossible to move forward without catering the failed task.

2.2.2. Proactive Fault Tolerance

The proactively replacement of suspected components with other healthy working components is the basic principal of proactive fault tolerance policy. The policy predicts the problem before it comes in the system. It avoids the recovery from 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 clean state.

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 feedback loop. The applications are continuously analyzed and monitors with this mechanism in the preemptive migration.

3. 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 are 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 the 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 are 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 absence of synchronization.

To measure the fault tolerance performance in 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.

4. 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 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 takes the decision of fault tolerance for the system. The reliability of nodes is adaptive and changes after every computing cycle. The reliability of virtual machine or the processing nodes increases if nodes produce the desired result with in the specified time limit, if not then the reliability decreases in the AFTRC model.

LLFT: The deployment 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 in a transparent manner 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 protect the applications against the various faults is provided by the middleware.

FTWS: Onthe basis of the priority of the tasks the replication and resubmission of tasks is carried out in fault tolerance work flow 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 modeling language (SysML) [15] is used to express the specifications of candy (component based availability modeling 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 **ranking** section to assign the ranks to components on the basis of calculated significant value of cloud components second **optimal fault tolerance** section 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 significant component.

Map-reduce: In 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 together

to achieved 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 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 fault and node as 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 the gossip architecture the replication policy exploits the fault detection and capability of fault tolerance in cloud computing environment. Every node selects a neighbor node hence two nodes have to update their decision vector at a time during the processing. The fault detection reliability in the gossip architecture increases 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 model for parallel programming uses the job migration and check point/restart techniques. The two layered structure of the MPI model in which upper layer infrastructure communication is not dependent and in lower layer it is specified that the backup of check point is required or not. The running job is migrated to the healthy node on receiving positive response from the faulty node using these techniques in this model.

5. Tools used for Implementing Fault Tolerance Techniques in Cloud Computing:

Fault tolerance challenges and techniques have been implemented using various tools. Followings are the major tools which are used to implement different fault tolerance techniques Based on their programming framework, environment and application type [**9**][**13**].

Amazon Elastic Compute Cloud: Amazon EC2 offers the scalable computing capacity and provides the facilities to launch and manages API based server instances using available tools [**21**]. It provides the ability to keep instance at different locations. Amazon provides the virtual computing environment, preconfigured packages that enable a user to run Linux based applications. It enables to build fault tolerant systems that work with low cost and minimum amount of human interaction. Amazon offers the secured services to the user on demand basis through the virtual private cloud.

Assure: Assure presents the rescue points which are the locations in existing application codes for handling programmer anticipated failure [**22**]. It can be used to provide alternative pathway which induces software and recover from software failures by using the error virtualization techniques to force an error return using an observed value in a function. Assure uses a production system and provide assistance on the basis of need of the system to implement rescue points and error virtualization.

Hadoop: Hadoop is open source java based software; provides a reliable and scalable framework for distributed computing. The two components, **HDFS** (Hadoop distributed file system) runs on node of server cluster to access data code and **Map Reduce Engine** to perform map reduce operations by dividing the input data into smaller pieces. The input data are replicated and stored at different machines to provide the fault tolerance via replication technique [**23**].

S-Help: S-help is a light weight automatic system that can survive software failure in virtual machine framework. S-help works as error handler in cloud computing environment. Initially zero as weight value is

assigned to each rescue point, and increased when a fault detected in the system [24]. The rolled back to the latest check point is decided by this rescue value.

HA –Proxy: It is referred as high-availability proxy. It is an open source software tool which provides a load balancing among large pool of web servers by distributing the load. It provides the fault tolerance capacity to the system. The Ha-Proxy handles the request by redirecting them to another server when a fault occurs in the system.

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
Check pointing	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
Check pointing, Retry, Self Healing	Reactive/ Proactive	Assure	JAVA	Virtual Machine	Host, Network Failure	Fault tolerance

Table 1: Tools Used for Fault Tolerance in Cloud Computing

6. Conclusion:

Cloud computing is the promising epitome for providing technical services as computing utilities. However, the remote location of the service provider, the user is susceptible about the reliability and dependability of the computing process. Hence, providing fault tolerance in a cloud computing system is a major research challenge. In this paper, major research challenges in providing fault tolerance in cloud computing systems along with fault tolerant techniques have been discussed. The techniques to achieve software as well as hardware fault tolerance

have been presented. The popular tools for implementing fault tolerant techniques in cloud environment have also been discussed and their comparison is summarized in a table 1.

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 perspective 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.