ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

A Review on Fault Tolerance Techniques in Cloud

Monika, Abhimanyu Jindal

Supervisor, Dept. of Information Technology, UIET, Panjab University, Chandigarh, India

Research Student, Dept. of Information technology, UIET, Panjab University, Chandigarh, India

ABSTRACT: As cloud computing is an emerging technology and people are moving towards cloud to use its services so it is important to provide fault free services to the cloud users. Fault tolerance is the property of the system in which the system can perform gracefully in case of hardware or software failure. Cloud is providing services to huge number of users from different locations with different access rights. In this paper, various fault tolerant techniques are discussed. These techniques make use of sensors, backups, various algorithms, redundant tasks and checkpointing to cop up with the faults in the cloud. A comparative analysis of these techniques on the basis of technology used by them has been shown with their pros and cons.

KEYWORDS: Cloud computing; Testing; Fault tolerance; Checkpointing; Mapreduce

I. INTRODUCTION

Testing is an essential part of every software development cycle. Without testing you cannot judge whether the calculated results are accurate or not. Each organization hires testing experts to test the software due to which it wastes time in recruiting the testing experts and it has to pay them for whole time. There are number of testing tools available in the market but there licensing costs are high. There is a growing trend to shift the testing tools onto the cloud infrastructure so that testing tools can be used according to pay per use policy. Organizations do not require to hire testing experts or purchase the testing tools due to which time and money are saved. So the demand for outsourcing testing is rapidly increasing. Cloud testing is an advantage for the companies who want testing but do not want to take risk of investment.

Cloud computing demands are increasing day by day as a result of which it is mandatory to provide accurate services in the presence of fault also. Fault tolerance is the procedure of finding faults in the system. Although there is hardware or software failure in the system yet the system should work properly. Fault tolerance has the property that it works gracefully with the system capabilities so that the system can work correctly against the hardware or software failure. In order to obtain robust cloud computing the failures should be handled with care. The considerable use of fault tolerance in cloud computing is to cope up with software and hardware failures, reducing cost and increasing the performance. A more reliable method than checkpointing can be used to cop up with VM failure. The service provider starts a new VM with same features as the failed VM and then it request the customer to deploy its application on this new VM.

Fault tolerance can be divided into two types according to the policies of fault tolerance: a) Proactive Fault Tolerance: Proactive fault tolerance means predicting the problem before it actually occurs, b) Reactive fault tolerance: It handles the failures. The effect of failure is minimized when it occurs.

The fault tolerance is required due to following reasons: a) The user defined nodes in cloud are cheaper, less powerful and less reliable, b) The communication faults like connection time out impact the reliability of cloud.

There are two distinct phases in fault tolerance in cloud: a) Detection phase and b) Repair phase. Then question arises on which cloud participant these phases should be implemented either on the cloud provider or the cloud customer. Cloud provider can detect and repair hardware failures, both cloud provider and customer can detect VM failures but only cloud provider can repair it and application failures can only be detected by cloud customer but both can repair it. In further sections of this paper, we will study about various approaches of fault tolerance used in various cloud architectures and will conclude the paper in section III.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

II. FAULT TOLERANCE TECHNIQUES IN CLOUD

A. *Fault Tolerance Using Sensors:*

Alain Tchana et al [2] discussed the implementation of fault tolerance on various levels:-At application level, fault detection is done using sensors. Sensors check the liveness of the running application and then attempt to repair the application using two methods i.e. stateless (restart the server on which fault has occured) and state-full(saves the state of application to restore it and save the work done before restarting the server).At cloud level,VM fault tolerance can be achieved by accessing VM hypervisor. Making use of hypervisors reduce the cost of the system as it reduces the no. of sensors to be used. Only single sensor can be used to monitor all the VMs on a machine. VM fault tolerance can also be achieved by checkpointing and then restarting the system from this saved state on failure.

B. *Collaborative fault tolerance:*

S. Girish et al [3] proposed a collaborative fault tolerance model that recovers the system from failures by using multiple copies of the input data. The model identifies and overcomes the frequently occurring cloud failures and maintain the integrity of the system. Hence it provides integrity, scalability and reliability.

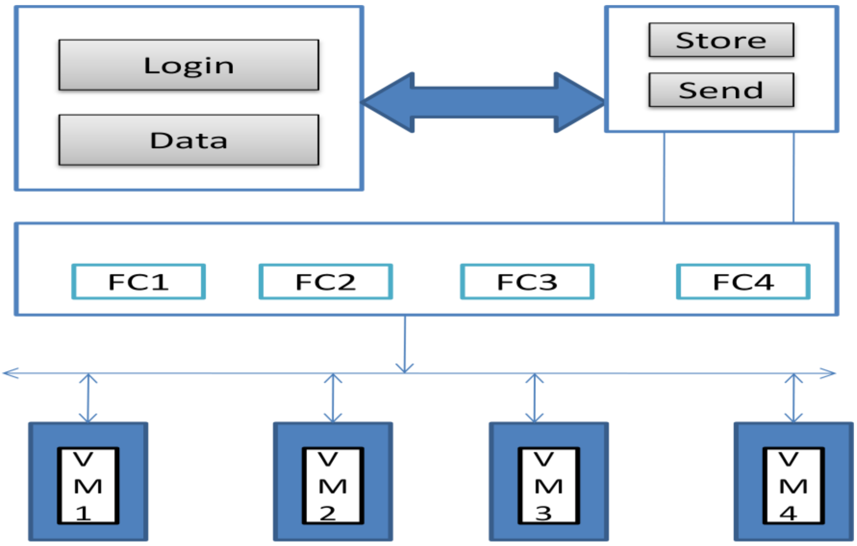

Fig.1. Sketch of collaborative fault tolerance model [3].

C. *MapReduce Fault Tolerance:*

Qin Zheng [4] made some improvements in the MapReduce system. The MapReduce system works in distributed environments where the tasks are executed on various machines of the system and the data is shared by all these machines. The data that is to be processed is divided into various blocks and is given to all the tasks on the various machines. In cloud, MapReduce fault tolerance is improved by providing redundancy for the tasks in system i.e. creating backups of the tasks. All the backup tasks created are allocated on various backup instances. Whenever a machine failure is encountered, the task is rescheduled on the instance.

D. *Fault Tolerance in Real-Time Cloud Computing:*

Sheheryar Malik and Fabrice Huet [5] proposed adaptive fault tolerance in real-time cloud computing. In this technique, on the basis of the reliability a virtual machine is selected for execution, if it fails to execute real time task the VM does not perform further real time applications. Different algorithms run on different virtual machines to check the reliability of the machine. The fault tolerance mechanism is applied to each VM. If the reliability of VM falls below the prefixed minimum reliability, the VM is declared as dead and is stopped to perform further operations.

E. *System-Level Approach to Fault Tolerance:*

Ravi Jhawar et al [6] provides a system level approach to fault-tolerance. This approach permits the users to specify and apply their desired level of fault tolerance without the knowledge of anything about its implementation. This approach inserts a dedicated service layer between the computing infrastructure and applications which consists of a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

replication manager(for creating Redundant copies of user applications), fault detection manager(detection of failures), fault masking manager(mask the failures), recovery manager, messaging monitor, client/admin interface, FTM kernel and resource manager. This service layer helps the system to achieve fault tolerance.

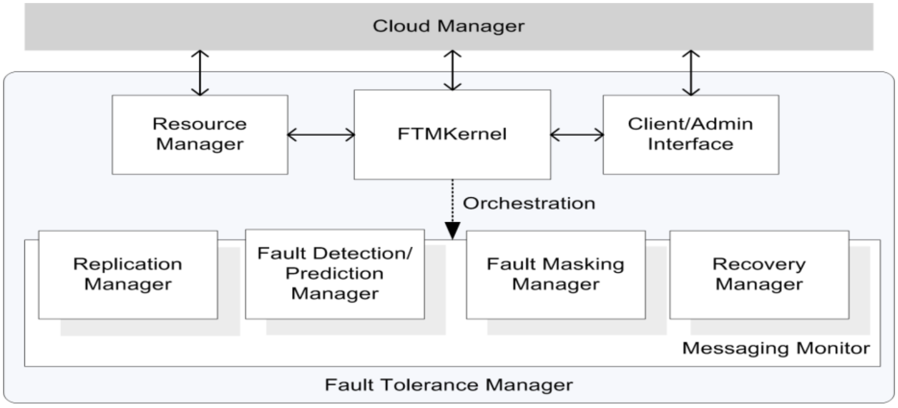

Fig.2. FTM Architecture [6].

F. *LLFT Framework:*

Wenbing Zhao et al [7] provided a Low Latency Falut Tolerant framework for implementing fault tolerance in the cloud. The processes in applications are in groups of two or more processes where one process is known as Primary and rest are backups. The LLFT protocol provides all the processes with messages containing the ordering information in group of messages and thus maintaining consistency among the replicas of processes.

G. *FTCloud:*

Zibin Zheng et al [8] discuss a component ranking framework for fault tolerance named FTCloud. It selects some significant components from the applications and then gives them rankings based on which a different FT strategy is applied to each component. The significance value for a component ci:

$$V(ci) = (1 - d)/n + d \sum_{k \in N(ci)} V(ck)W(eki),$$

where, n is the no. of components, N(ci) is a set of components that invoke component ci and W(eki) is the weight value.

H. *BFTCloud:*

Yilei Zhang et al [9] present a BFTCloud (Byzantine Fault Tolerant Cloud) framework which could guarantee robustness of the system. The framework consisted of 257 resource providers located in 26 countries and claim robustness of system when up to *f* of total *3f+1* resource providers are faulty. The framework basically makes use of replicas for implementation of fault-tolerance.

Later on Giuliana Santos Veronese et al [10] claim that same performance of the BFT framework can be achieved by using nonly 2f+1 replicas and it also reduce the cost of the system.

Peter Garraghan et al [11] also developed a framework called FT-FC (Fault-Tolerant Federated-Cloud) which helps in creation of Byzantine FT systems and apply them to cloud systems inorder to achieve fault tolerance.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

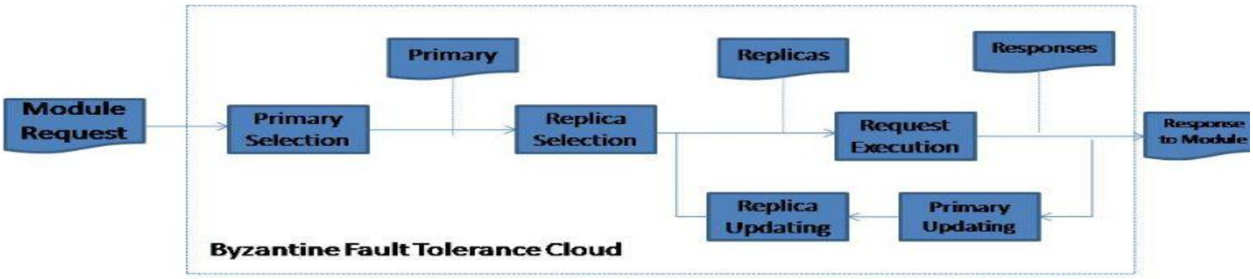

Fig.3Byzantine Fault Tolerance Cloud[10]

I. *Assure:*

Stelios Sidiroglou et al [12] presented a system Assure that uses the concept of rescue points for recovering the application from faults. The rescue points are identified using fuzzy and then are implemented using checkpoint and resume strategy.

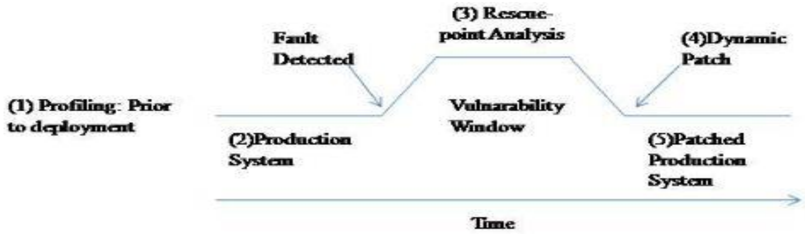

Fig 4.Assure[12]

J. *Checkpointing:*

Lin and Ahmed [13] also discussed a checkpoint mechanism where the system takes the checkpoints of tasks periodically and saves these checkpoints on a non-volatile storage memory. Whenever a failure occurs, the system rolls back the tasks to the most recent checkpoint and then start the re-executing of these tasks.

K. *DDFMCS:*

Liying Wu et al [14] put forward a dynamic data fault tolerance for cloud storage (DDFMCS). This strategy dynamically determines the FT mechanism for the application using the metadata of the file stored in its file tables.

L. *SFD:*

Naixue Xiong et al [15] proposed a Self-tuning Failure Detector (SFD) which could adjust control parameters dynamically to achieve fault tolerance which is an improvement over the existing FT schemes which don't adjust the parameters on run-time.

M. *Delay Tolerant FT Algorithm:*

Jameela Al-Jaroodi et al [16] proposed a delay- tolerant FT algorithm which reduces execution time of applications and adapts to the failures. This algorithm is mainly effective in downloading and execution of multiple parallel applications on various servers in cloud.

N. *Dynamic Power and failure Repair:*

Sampaio and Barbosa [17] proposed some algorithms for mapping of virtual machines to physical hosts to improve the power-efficiency of cloud. The algorithms make use of proactive FT technique to deal with the failures.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

O. *Unibus:*

Magdalena Slawinska et al [18] proposed the use of Unibus (a project that focus on resource access and provisioning) to employ fault- tolerance in cloud by making use of DMTCP (Distributed MultiThreaded Checkpointing).

P. *VFT:*

Das and Khilar [19] proposed a Virtualization and Fault Tolerance (VFT) technique. The technique uses Cloud Manager (CM) and Decision Maker(DM) modules for performing virtualization, load balancing and then fault-tolerance. Fault-tolerance is achieved using redundancy, checkpointing and fault handler.

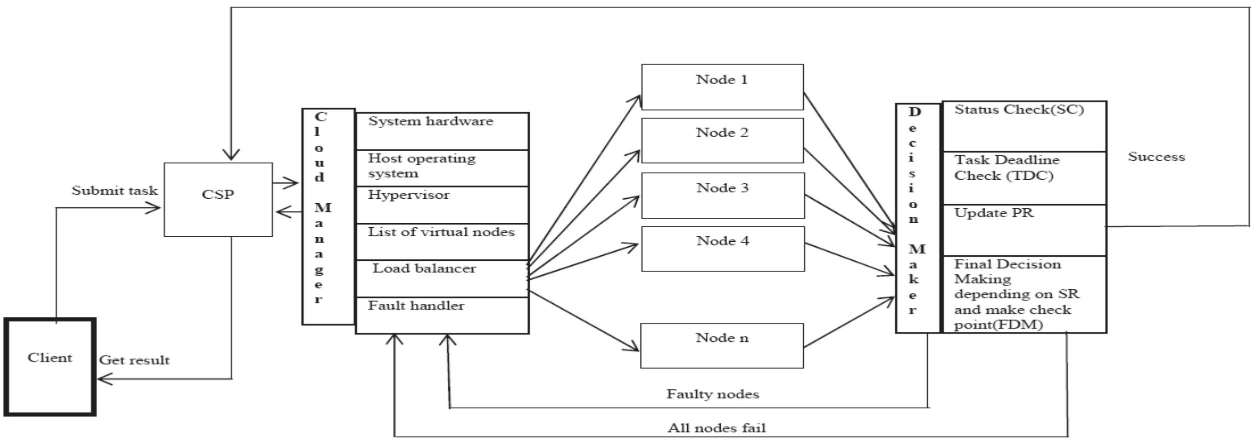

Fig.6. VFT Model [19]

Table 1: Comparison of FT techniques

S.No.	Name of the Technique	Technology employed/Strategy	Pros	Cons
1	Fault tolerance using sensors	Sensors, Checkpointing	Sensors made detection of faults easy	A lot of sensors are required which increases cost, Checkpointing saves entire VM state instead of difference between states.
2	Collaborative fault tolerance	Data Duplication	User authentication, Data recovery	Works greatly on small sized networks only
3	MapReduce fault tolerance	Backups of tasks	Contain backup of all tasks	No data locality
4	Adaptive fault tolerance	Reliability Algorithms	Early detection of VM reliability, Pocess is insensitive to domino effect	Min. and max. values of reliability are not calculated
5	System level approach to fault tolerance	A service layer	Low cost, Easy to use	Overhead on system developer due to prediction of user requirements
6	LLFT	Messaging protocol	Low end-to-end latency, replica consistency	Cost and overhead of maintaining replicas.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

7	FTCloud	Ranking protocol	Efficient ranking of components	Don't consider latency, Component failure correlations.
8	BFTCloud	Multiple copies	Guarantees robustness of system	Overhead of replicas
9	Assure	Rescue points	Efficient recovery on failure	Overhead of maintaining rescue points
10	DDFMCS	Metadata, stored files	Dynamic	Not efficient for large tables
11	SFD	Self-tuning failure detector	Dynamic adjustment of control parameters	Difficult to implement
12	Delay tolerant algorithm	Various algorithms for task division	Reduction in execution time, delay.	Various servers are required.
13	Unibus	Unibus system	Checkpointing, Fast recovery	Overhead of saving large number of checkpoints.
14	VFT	Cloud manager and decision manager module	Virtualization, load balancing	Overhead of saving large number of checkpoints.

III. Conclusion and Future Work

Fault tolerance is an essential aspect of a cloud as most of the applications fail due to occurrence of faults during their execution. Due to fault, the running time of the application increases and resources of the cloud are wasted. So, an efficient technique to cop up with the faults is required in the cloud. All the techniques discussed here are highly efficient in detecting faults and recovering the system from it.

Application of sensors make detection of the faults easy but it increases the cost of using sensors and increases the complexity of the system, so we can use only single sensor for a complete machine. Checkpointing saves the time and resources for restarting the application but it needs a lot of space and time for saving the states of all tasks again and again and thus it can be improved by saving only the difference between successive VM states. Backups also serve their working by assisting the tasks during their execution but again creating backups may require extra efforts of the service provider. In future, analysis of which fault tolerant technique should have priority to compute the fault tolerance will be done that will help the clouds to recover the failed VM in lesser time.

References

1. Deepali Mittal and Neha Agarwal, "A review paper on fault tolerance in cloud computing," International Conference on Computing for Sustainable Development(INDIACom), IEEE, 2015, pp.31-34.
2. A. Tchana, L. Broto and D.Hagimount," Approaches to cloud computing fault tolerance," International Conference on Computer, Information and Telecommunication Systems(CITS), IEEE, 2012, pp.1-6.
3. S. Giriesh, V. Sindhuja, P. Padmakumari and A. Umamakeswari,"Dynamic data fault tolerance mechanism to enhance reliability and availability in cloud," Indian Journal of Science and Technology, Vol 8(S9), 2015, pp. 300-305.
4. Qin Zheng,"Improving mapreduce fault tolerance in cloud," IEEE International Symposium on Parallel and Distributed Processing(IPDPSW), IEEE, 2010, pp. 1-6.
5. Sheheryar Malik, Fabrice Huet,"Adaptive fault tolerance in real time cloud computing," IEEE World Congress on services(SERVICES), IEEE, 2011, pp.280-287.
6. R. Jhawar, V. Piuri, Macro Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in cloud computing," IEEE International Systems Conference(SysCon), IEEE, 2012, pp.1-5.
7. Wenbing Zhao, Melliar-Smith, and P. M. Moser, "Fault tolerance middleware for cloud computing," in 3rd International Conference on Cloud Computing (CLOUD 2010), Miami, FL, USA, 2010.
8. Z. Zheng et al, "FTCloud: A component ranking framework for fault-tolerant cloud applications," 21st International Symposium on Software Reliability Engineering, IEEE, 2010, pp.398-407.
9. Y. Zhang, Z. Zheng, and M. R. Lyu. "BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing," International Conference on Cloud Computing (CLOUD), IEEE, 2011, pp. 444-451.
10. G.S. Veronese et al, "Efficient byzantine fault tolerance," IEEE transactions on computers," IEEE, 2013, pp.16-30.
11. P. Garraghan, P. Townend and J. Xu, "Byzantine fault-tolerance in federated cloud computing," 6th International Symposium on Service Oriented System Engineering, IEEE, 2011, pp.280-285.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

12. S. Sidiroglou et al, "Assure: Automatic software self-healing using rescue points," In ACM Sigplan notices, 2009, pp.37-48.
13. Luke lin,Mustaque ahamad,"Checkpointing and rollback recovery in distributed object based system," 20th International Symposium, IEEE, 1990, pp. 97-104.
14. L.Wu, Bo Liu, W.Lin, "A dynamic data fault-tolerance mechanism for cloud staorage," 4th International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, 2013, pp.95-99.
15. N. Xiong et al, "A self-tuning failure detection scheme for cloud computing service," 26th International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2012, pp.668-679.
16. J. Al-Jaroodi et al, "An efficient fault tolerant algorithm for distributed cloud services," 2nd Symposium on Network Cloud Computing and Applications," IEEE, 2012, pp.1-8.
17. A.M.Sampaio and J.G.Barbosa, "Dynamic power and failure aware cloud resources allocation for sets of independent tasks," IEEE International Conference on Cloud Engineering, IEEE, 2013, pp.1-10.
18. M. Slawinska, J. Slawinski and V. Sunderam, "Unibus: Aspects of heterogeneity and fault tolerance in cloud," International Symposium on Parallel and Distributed Processing, Workshops and Phd Forums (IPDPSW), IEEE, 2010, pp.1-10.
19. P. Das, P.M. Khilar, "VFT: A Virtualization and Fault Tolerance Approach for Cloud Computing," IEEE Conference on Information & Telecommunication Technologies, IEEE, 2013, pp.473-478.