Fault Tolerance in Cloud Computing – Survey

Salma M. A. Ataallah, Prof. Salwa M. Nassar
Computers and Systems Department
Electronics Research Institute - ERI
Giza, Egypt
salma_mohamed@eri.sci.eg, salwa@eri.sci.eg

Prof. Elsayed E. Hemayed
Computer Engineering Department
Faculty of Engineering, Cairo University
Giza, Egypt
hemayed@ieee.org

Abstract—**Cloud computing datacenter hosts hundreds of thousands of servers that coordinate users' tasks in order to deliver highly available computing service. These servers consist of multiple memory modules, network cards, storage disks, processors etc…, each of these components while capable of failing. At such a large scale, hardware component failure is the norm rather than an exception.**

Hardware failure can lead to performance degradation to users and can result in losses to the business. Fault tolerant is one of efficient modules that keep hardware in operational mode as much as possible. In this paper, we survey the most famous fault tolerance technique in cloud computing, and list numerous FT methods proposed by the research experts in this field.

Key words—**Cloud Computing, Fault Tolerance.**

I. Introduction

Cloud computing is a type of computing that relies on sharing a pool of computing resources, rather than deploying local or personal hardware and software. According to National Institute of Standards and Technology, USA. (NIST), *"Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

Due to the increasing of resources demand in cloud, flexibility in obtaining and releasing resources is an important issue in the cloud. The availability of extending resources pool for the user provides an effective alternative to deploy applications with high scalability and processing requirements. In general, a Cloud computing infrastructure is built by interconnecting large-scale data centers, and computing resources are delivered to the user over the Internet in the form of an on-demand service by using virtual machines.

Cloud computing has multiple characteristics; high availability and reliability are most essential cloud characteristics, the traditional way for achieving reliable and highly available cloud service is to use a fault tolerance module. The simplest definition of fault tolerant system is the ability to continue computing processing despite the presence of a hardware failure.

The failures that occur in cloud computing can be classified into two classes namely; First, Data Failures such as data corruption, missing source data and other flaws in the data. Second, Hardware failure such as faulty or slow VMs and storage access exception.

The rest of the paper is organized as follows: The next section presents fault tolerance requirements that are used to evaluate any fault tolerant mechanism. In section III, we present different fault tolerance policies in the cloud. Finally, in section IV we summarize different developed fault tolerance techniques.

II. Fault Tolerance Requirements

An important goal in designing distributed systems is constructing the system in such a way that it can automatically recover from partial failures without seriously affecting the overall performance. In particular, whenever a failure occurs, the system should continue operating in an acceptable way while repairs are being made. In other words, a distributed system is expected to be fault tolerant.

To understand the role of fault tolerance, we should cover a number of useful requirements for distributed systems including the following:

- **Mean Time To Failure (MTTF):** It is the expected time to failure given that the system has been operational up.
- **Mean Time To Repair (MTTR):** It refers to the expected time to repair the system after failure occurrence.
- **Mean Time Between Failures (MTBF):** It represents the average time to the next failure, and is calculated as:

$$\text{MTBF} = MTTF + MTTR \qquad (1)$$

978-1-5090-0275-7/15/$31.00 ©2015 IEEE

- **Reliability*:*** It refers to the property that a system can run continuously without failure.

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

- **Availability:** It refers to the probability that the system is operating correctly at any given moment and is available to perform users' tasks.

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

- **Safety:** It refers to the situation that when a system temporarily fails to operate correctly, nothing catastrophic happens.

- **Maintainability:** It refers to how easy a failed system can be repaired. A highly maintainable system may also show a high degree of availability.

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

Fault tolerance recognizes that faults may exist; the main role of fault tolerance is to try concealing the occurrence of system failures. There are two phases of fault tolerance, these are: ***Error detection*** which provides hints about the operational status of processes. ***Error Recovery*** that attempts to transform the erroneous system state into an error free state.

The key technique for faults tolerant is to use redundancy. In redundancy, extra equipment or processes are added to make it possible for the system as a whole to tolerate the loss of some processes.

III. Cloud Fault Tolerance Policies

In Cloud Computing, fault tolerance is the ability of the Cloud to withstand the unexpected changes which occur due to hardware faults, software faults, network congestions etc.

There are mainly two standard fault tolerant policies available for real-time applications hosted in cloud namely Proactive Fault Tolerance Policy and Reactive Fault Tolerant Policy [1][8].

Proactive Fault Tolerance:

The proactive fault tolerance principle is to avoid recovery from faults, errors and predict the failure and proactively replace the suspected components from other working components. Some of the techniques based on proactive fault tolerance policies are:

1) Preemptive migration:

It is a technique in which feedback-loop control system is used where applications are constantly monitored and analyzed.

2) Software Rejuvenation:

It refers to schedule periodic reboots for the system. After each reboot, the system resumes with a clean state.

Reactive Fault Tolerance:

In Reactive fault tolerance policy, it deals with measures which are applied to reduce the effect of the faults already occurred in Cloud.

There are various techniques which are based on reactive fault tolerance policy like Checkpointing/Restart, Replication, and Task Resubmission.

1) Check pointing/Restart:

In Checkpointing/Restart mechanism, the state of a system -that is running an application- is recorded in a global checkpoint. So in the event of a fault, the system state can be rolled back to the checkpoint and allowed to continue from that point, rather restarting the application from the beginning.

2) Replication:

Replication based technique is one of the popular fault tolerance techniques. Replication is a process of maintaining different copies of a data item or object on different resources. Replication adds redundancy in the system.

3) Task Resubmission:

At fault detection, the task is submitted either to the same or to a different resource at a runtime without interrupting the workflow of the system.

IV. Cloud Fault Tolerance Developed Techniques

In last year’s, a lot of cloud computing communities developed various fault tolerance techniques to improve their cloud performance. In this section, we introduce some of these.

A Proactive Fault Tolerance:

A Proactive Fault Tolerance (FT) [9] is a fault tolerance approach to High Performance Computing (HPC) in the cloud that is used to reduce the wall clock execution time in the presence of faults. It does not rely on a spare node prior to prediction of a failure. Proactive FT uses an avoidance mechanism to tolerate faults. It achieves this by relying on the system log and health monitoring facilities.

Fault Tolerance Manager (FTM):

Fault Tolerance Manager (FTM) is an extra service layer which offers required fault tolerance properties to the applications as an on-demand service [10] [11] [12]. Authors present a failure model for cloud infrastructures such as server components (including VM and VMM), network and power distribution, to analyze the impact of each failure on user's applications. Also they introduced an innovative, system-level, modular perspective on creating and managing fault tolerance in Cloud computing environment.

SHelp: Automatic self-healing of server applications in a virtual machine environment:

SHelp [13] is an error handler for the same application instances which are run in different VMs hosted on one physical machine. SHelp uses the Berkeley Lab Checkpoint/Restart (BLCR) [14] as the checkpoint and rollback tool. Authors introduced two new techniques, namely, **weighted rescue points** and **two-level rescue point database**.

In weighted rescue point, each rescue point is assigned a weight value which is initially set to zero; the associated weight value is incremented each time a fault is found in a function. When a fault occurs, the application is rolled back to a latest checkpoint, and first uses error virtualization at a rescue point which has the largest weight value among the candidate rescue points, then at the rescue point with the second largest weight value and so on until the fault is bypassed.

In two-level rescue point database, the fault related information could be shared among applications in different virtual machines to enable applications to recover from the future faults more quickly.

A Self-tuning Failure Detection (SFD):

SFD [15] is a dynamic failure detector scheme; SFD can adjust well to handle unexpected network conditions and the requirements of any number of concurrently running applications. Authors carried out actual and extensive experiments to compare the quality of service performance between the SFD and several other existing FDs. One of SFD properties is appropriate for the "one monitors multiple" and "multiple monitor multiple" cases based on the parallel theory.

An Efficient Fault-Tolerant Algorithm:

An Efficient Fault-Tolerant Algorithm [16] proposes a delay-tolerant fault tolerance algorithm which adapts failures by effectively reducing execution time and thus minimizing the fault discovery & recovery overhead in the Cloud. The algorithm claims to be used efficiently in places like cloud which handles distributed tasks. According to them, the algorithm ensures that data gets downloaded reliably from replicated servers and efficiently executing applications on independent multiple distributed servers in the Cloud.

A Byzantine Fault Tolerance Framework (BFTCloud):

BFTCloud [17] is a fault tolerance scheme that is used for building robust systems in voluntary-resource cloud environments. BFTCloud uses replication techniques for overcoming failures since a broad pool of nodes are available in the cloud; BFTCloud select voluntary nodes based on their QoS characteristics and reliability performance. Faulty voluntary resources will be replaced with other suitable resources once they are identified.

A cloud computing fault net (CFN):

CFN [18] is used to precisely model the different components of cloud computing, such as service resources, cloud module, the detection and failure process …etc. Petri net is used to create the different components of Cloud Computing which gets integrated dynamically into CFN model. Based on CFN model, the properties of the components are analyzed developing a fault detection strategy at each level which dynamically detects the faults in the execution process.

Virtualization and Fault Tolerance (VFT):

VFT [19] is a reactive fault tolerant technique; it consists of a Cloud Manager (CM) module and a Decision Maker (DM) which are used to manage the virtualization, load balancing and to handle the faults. The first step involves virtualization & load balancing and in the second step fault tolerance is achieved by redundancy, checkpointing and fault handler. The virtualization includes a fault hander. Not all the faults are recoverable. Fault handler finds these unrecoverable faulty nodes and restricts these virtual nodes from future requests or usage. It also helps to remove the temporary software faults from recoverable nodes making them available for future requests.

Adaptive Fault Tolerance in Real-time Cloud computing (AFTRC):

AFTRC [20] is a fault tolerance model for real time Cloud Computing. In this model, the faults are managed based on the reliability of processing nodes or virtual machine. According to authors, the reliability of nodes changes in every computational cycle.

In table 1, we summarize previous techniques and compare them by the following attributes:

Resource awareness indicates if this technique depends on resources or it can be used with any resources. Distributed Environment represents which environment is used for development. Polices indicates which policy is used. Reliability & Availability are the most used fault tolerance requirements. Finally, the last attribute is the programming framework.

Table 1: Tools Used For Implementation of Fault Tolerance Techniques

Fault Tolerance Techniques	Resource Awareness	Distributed Environment	Policies	Reliability	Availability	Programming Framework
A Proactive Fault Tolerance [9]	Yes	HPC on Cloud	Proactive	N/A	Yes	Java
FTM [10][11][12]	Yes	Cloud	Reactive	N/A	N/A	N/A
SHelp [13]	Yes	Virtual Machine	Reactive	No	No	C
SFD [15]	No	Cloud Networks	Reactive	N/A	N/A	Java
An Efficient Fault-Tolerant Algorithm [16]	No	Cloud	Reactive	N/A	N/A	N/A
BFTCloud [17]	No	Cloud	Reactive	No	Yes	Java
CFN [18]	No	Cloud	Reactive	N/A	N/A	Petri net
VFT [19]	No	Cloud Virtualization	Reactive	No	No	N/A
AFTRC [20]	No	Cloud	Proactive/ reactive	Yes	N/A	N/A

V. Conclusion

Over the latest years, Cloud Computing has become a popular computational technology across all industries. Cloud has vast advantages like providing access to large amount of data & resources, on-demand service provisioning, reduced cost of managing the infrastructure etc. making it unique from other technologies. Reliability and availability of the services are of the most important cloud services. So there is a need for an efficient fault tolerance method which shields the Cloud from faults or failures. In this paper, we concentrate on the standard fault tolerant concepts in Cloud Computing. Because Cloud Computing is a new field of research compared to other technologies, lot of research works are being carried out, especially in developing a standalone fault tolerance method. There are numerous FT methods proposed by the research experts in this field.

References

[1] A. Ganesh, M. Sandhya, S. Shankar. "A Study on Fault Tolerance methods in Cloud Computing." In International Advance Computing Conference (IACC), 2014 IEEE International Conference on, pp. 844-849. IEEE, 2014.

[2] Peter Mell and Timothy Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication 800-145, Sept. 2011.

[3] K. V. Vishwanath and N. Nagappan, “Characterizing cloud computing hardware reliability,” in Proceedings of the 1st ACM symposium on Cloud computing, ser. SoCC '10. New York, NY, USA: ACM, 2010, pp. 193– 204.

[4] Forbes, R. E., Rutherford, D. H., Streglitz, C. B., and Tung, L. H., "A Self-Diagnosable Computer,"

AFIPS Fall Joint Computer Conf. Proc., vol. 27, pt. 1, 1073-1086, 1965.

[5] B. Randell, "System structure for software fault tolerance," In IEEE Transactions on Software Engineering SE-1, pp. 220-232 (June 1975).

[6] J. J. Horning, H. C. Lauer, P. M. Melliar-Smith, and B. Randell, “A program structure for error detection and recovery,” in Proc. Conf. Operating Systems: Theoretical and Practical Aspects, IRIA, Apr. 23-25, 1974, pp. 177-193.

[7] Somani A. K. and Vaidya N. H., "Understanding Fault Tolerance and Reliability," IEEE Computer, pp. 45-46, April 1997.

[8] K. Ganga., and S. Karthik. "A fault tolerent approach in scientific workflow systems based on cloud computing." In Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference on, pp. 387-390. IEEE, 2013.

[9] I. P. Egwutuoha, S. Chen; D. Levy, Bran Selic and R. Calvo, "A Proactive Fault Tolerance Approach to High Performance Computing (HPC) in the Cloud," in The 2nd International Conference on Cloud and Green Computing, Xiangtan, Hunan, China, 2012, pp. 268-273.

[10] R. Jhawar, V. Piuri, and M. D. Santambrogio, "A comprehensive conceptual system-level approach to fault tolerance in cloud computing." In Systems Conference (SysCon), 2012 IEEE International, pp. 1-5. IEEE, Mar 2012.

[11] R. Jhawar, V. Piuri, and M. D. Santambrogio, "Fault tolerance management in IaaS clouds." In Satellite Telecommunications (ESTEL), 2012 IEEE 1st AESS European Conference on, pp. 1-6. IEEE, 2012.

[12] R. Jhawar, V. Piuri, and M. D. Santambrogio, "Fault Tolerance Management in Cloud Computing: A System-Level Perspective," Systems Journal, IEEE , vol.7, no.2, pp.288-297, June 2013.

[13] G. Chen,, H. Jin, D. Zou, B. B. Zhou, W. Qiang, and G. Hu. "SHelp: Automatic Self-healing for Multiple Application Instances in a Virtual Machine Environment." In Cluster Computing (CLUSTER), 2010 IEEE International Conference on, pp. 97-106. IEEE, 2010.

[14] P. Hargrove and J. Duell, “Berkeley lab checkpoint/restart (BLCR) for Linux clusters”, Proceedings of the 2nd Scientific Discovery through Advanced Computing Program Conference (SciDAC’06), June 25-29, 2006, Denver, USA.

[15] N. Xiong, A. V. Vasilakos, J. Wu, Y. R. Yang, A. Rindos, Y. Zhou, W. Z. Song, and Y. Pan. "A self-tuning failure detection scheme for cloud computing service." In Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International, pp. 668-679. IEEE, 2012.

[16] J. Al-Jaroodi, N. Mohamed, and K. Al Nuaimi. "An Efficient Fault-Tolerant Algorithm for Distributed Cloud Services." In NCCA, pp. 1-8. IEEE, 2012.

[17] Y. Zhang, Z. Zheng, and M. R. Lyu. "BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing." In Cloud Computing (CLOUD), 2011 IEEE International Conference on, pp. 444-451. IEEE, 2011.

[18] G. Fan, H. Yu, L. Chen, and D. Liu. "Model Based Byzantine Fault Detection Technique for Cloud Computing." In Services Computing Conference (APSCC), 2012 IEEE Asia-Pacific, pp. 249-256. IEEE, 2012.

[19] P. Das, and P. M. Khilar. "VFT: A virtualization and fault tolerance approach for cloud computing." In Information & Communication Technologies (ICT), 2013 IEEE Conference on, pp. 473-478. IEEE, 2013.

[20] S. Malik, and F. Huet. "Adaptive Fault Tolerance in Real Time Cloud Computing." In Services (SERVICES), 2011 IEEE World Congress on, pp. 280-287. IEEE, 2011.