International Journal of Database Theory and Application
Vol.8, No.4 (2015), pp.183-190
http://dx.doi.org/10.14257/ijdta.2015.8.4.19

Fault-Tolerance Techniques in Cloud Storage: A Survey

Seyyed Mansur Hosseini and Mostafa Ghobaei Arani

Department of Computer Engineering, Islamic Azad University of Mahallat, Mahallat, Iran
Department of ComputerEngineering, Islamic Azad University of Parand, Tehran, Iran
,mostafaghobaei@piau.ac.irmansourh64@gmail.com

Abstract

In recent years, cloud computing is highly embraced and more organizations consider at least some type of cloud strategy and apply theming their business process. Since failure is probable in cloud data centers and access to cloud resources available is fundamental, evaluation and application of different fault-tolerance methods is inevitable. On the other hand, the increasing growth of cloud storage users motivated us to study fault-tolerance techniques, and their strengths and weaknesses. In this paper, after introducing the concept off ault-tolerance in the context of cloud computing, the fault-tolerant techniques are presented, and after introduction of some measures, a comparative analysis is provided.

***Keywords**: Cloud computing, Fault-Tolerance Technique, Cloud Storage*

1. Introduction

"Cloud computing" as a new paradigm in widely distributed computing, embraces cyber infrastructure built on the concepts of virtualization, grid computing, utility computing, network, web service, and software service to implement a service-oriented architecture[1,2].In order to reduce overall IT costs for the end user, a shared pool of computing resources is required to provide high flexibility and reduce total cost of ownership in the high-demand services [3,4].Cloud computing has the ability to connect the Internet and extensive network in order to use resources that are available remotely, and as a result, present efficient solutions based on pay-per-use [5, 6].Due to the exponential and fast growth of cloud computing, the need for fault tolerance in the cloud is a key factor for consideration.

Fault-tolerance, includes all the techniques necessary for robustness and dependability. The main advantages of using fault tolerance in cloud computing includes failure recovery, lower costs, and improved standards in performance [7].It is clear that cloud computing has become a center to run distributed applications by exploiting the different layers of virtualization, and can provide more flexibility in designing applications [8].The dependability of some aspects of the QoS provided by the system consists of features such as dependability and availability [9].

Motivation to study fault-tolerance techniques and models in the cloud computing has encouraged researchers to participate in the development of more efficient algorithms. Fault tolerance is an important aspect in cloud storage, due to the strength of the stored data [10,11].Although sufficient research has been done on cloud file system storage [10-14], deep studies are not carried out on the comparison of the fault-tolerance mechanisms used in cloud storage. This article will explain the various aspects of the faults and the need for fault-tolerance in cloud computing. The rest of this article is organized as follows: The second section describes classification of faults and the need for fault-tolerance in cloud computing. The third section discusses the fault-tolerance techniques in cloud computing and, provides a comparison of them. The fourth section examines criteria for fault-tolerance in cloud computing, and finally, the fifth section presents the conclusion.

2. Fault Tolerance Techniques

2.1. Fault Classification and the Need for Fault Tolerance in Cloud Computing

Increased demand for resources has led to an increase in the services, and thus establishment of large-scale data centers. In the past, high performance was considered as the main criteria in the design of data centers. Today, with the development of cloud computing-based data storage centers and demand for the use of cloud services, failure is common in today's data centers, which can be attributed partly to the large scales of data stored. As the scale data raises, access to them gets more complicated, so that different levels of access may be required for each application or each data item. The purpose of fault tolerance is to achieve robustness and dependability in every system. Based on the policies and procedures of fault tolerance, the methods can be divided into two categories: proactive and reactive.

Proactive fault tolerance policy prevents retrieval of fault, error, and failure with the help of prediction, and proactive fault tolerance detects the suspicious items and replaces it with the correct data, which means that I discover the problem before it really occurs.

Reactive fault-tolerance policy tries to reduce failures when they occur. It can be divided into error processing and fault-treatment techniques. The purpose of error processing is to eliminate errors from the calculation. Error treatment also aims to prevent the reactivation of errors [13, 15].

Fault-tolerance by error processing, which is composed of two phases, is performed as following: The "effective error processing" stages, that intends to hide effective fault before occurrence of failure, and "hidden error processing" that aims to ensure that the faults will not be activated again [16].

2.2. Existing Fault Tolerance Techniques in Cloud Computing

A large number of studies have conducted on the data fault-tolerant systems in the recent years that caused the bringing up the new strategies for finding the benefits and obstacles of fault-tolerant systems. In this section, we introduce the most recent fault-tolerance techniques in the Cloud Computation [7, 8, 14, 17, 18].

Check pointing–It is an efficient task level fault-tolerance technique for long running and big applications .In this scenario after doing every change in system a check pointing is done. When a task fails, rather than from the beginning it is allowed to be restarted that job from the recently checked pointed state.

Job Migration –Some time it happened that due to some reason a job can- not be completely executed on a particular machine. At the time of failure of any task, task can be migrated to another machine. Using HA-Proxy job migration can be implemented.

Replication- It is one of the most significant fault-tolerant techniques in storage centers that widely used in laboratory settings and in online service systems. Replication means “to copy.” Different tasks are replicated for successful execution and optimal results, so replication performs on different resources. Replication can be executed through HA-Proxy, Hadoop and AmazonEC2.

Self-Healing- A big task can divided into parts .This Multiplications done for better performance. When various instances of an application are running on various virtual machines, it automatically handles failure of application instances.

Safety-bag checks: In this case the blocking of commands is done which are not meeting the safety properties [8].

S-Guard- It is less turbulent to normal stream processing. S-Guard is based on rollback recovery. S-Guard can be implemented in Hadoop, Amazon EC2.

Retry- In this case we implement a task again and gain. It is the simplest technique that retries the failed task on the same resource.

Task Resubmission- A job may fail now whenever a failed task is detected, In this case at runtime the task is resubmitted either to the same or to a different resource for execution.

Timing check: This is done by watch dog. This is a supervision technique with time of critical function [8].

Rescue workflow- This technique allows the workflow to persist until it becomes unimaginable to move forward without catering the failed task.

Software Rejuvenation-It is a technique that designs the system for periodic reboots. It restarts the system with clean state and helps to fresh start.

Preemptive Migration- Preemptive Migration count on a feedback-loop control mechanism. The application is constantly monitored and analyzed.

Masking: After employment of error recovery the new state needs to be identified as a transformed state. Now if this process applied systematically even in the absence of effective error provide the user error masking [10].

Reconfiguration: In this procedure we eliminate the fault component from the system.

Resource Co-allocation: This is the process of allocating resources for further execution of task.

User specific (defined) exception handling- In this case user defines the particular treatment for a task on its failure.

2.3. Fault-tolerance Models in Cloud Computing

Currently, the various models based on the mentioned techniques in section 2.2.

AFTRC: a fault-tolerance model for real time cloud computing based on the fact that a real time system can take advantage the computing capacity, and scalable virtualized environment of cloud computing for better implement of real time application. In this proposed model the system tolerates the fault proactively and makes the diction on the basis of reliability of the processing nodes [15].

LLFT: is a propose model which contains a low latency fault-tolerance (LLFT) middleware for providing fault-tolerance for distributed applications deployed with in the cloud computing environment as a service offered by the owners of the cloud. This model is based on the fact that one of the main challenges of cloud computing is to ensure that the application which are running on the cloud without a hiatus in the service they provided to the user. This middleware replicates application by the using of semi-active replication or semi-passive replication process to protect the application against various types of faults [19].

FTWS: is a proposed model which contains a fault-tolerant work flow scheduling algorithm for providing fault-tolerance by using replication and resubmission of tasks based on the priority of the tasks in a heuristic matric. This model is based on the fact that work flow is a set of tasks processed in some order based on data and control dependency. Scheduling the workflow included with the task failure consideration in a cloud environment is very challenging. FTWS replicates and schedule the tasks to meet the deadline [20].

FTM: is a proposed model to overcome the limitation of existing methodologies of the on-demand service. To achieve the reliability and resilience they propose an innovative perspective on creating and managing fault-tolerance .By this particular methodology user can specify and apply the desire level of fault-tolerance without requiring any knowledge about its implementation. FTM architecture this can primarily be viewed as an assemblage of several web services components, each with a specific functionality [21].

CANDY: is a component base availability modeling frame work, which constructs a comprehensive availability model semi automatically from system specification describe by systems modeling language. This model is based on the fact that high availability assurance of cloud service is one of the main characteristic of cloud service and also one of the main critical and challenging issues for cloud service provider [22].

Vega-warden: is a uniform user management system which supplies a global user space for different virtual infrastructure and application services in cloud computing environment. This model is constructed for virtual cluster base cloud computing environment to overcome the 2 problems: usability and security arise from sharing of infrastructure [23].

FT-Cloud: Are a component ranking based frame work and its architecture for building cloud application. FT-Cloud employs the component invocation structure and frequency for identify the component. There is an algorithm to automatically determine fault-tolerance stately [24].

MAGI-CUBE: a high reliable and low redundancy storage architecture for cloud computing. The build the system on the top of HDFS and use it as a storage system for file read /write and metadata management. They also built a file scripting and repair component to work in the back ground independently. This model based on the fact that high reliability and performance and low cost (space) are the 3 conflicting component of storage system. To provide these facilities to a particular model Magi cube is proposed [25].

Table 1. Comparison of Various Models based on Protection against the Type of Fault and Procedure

Model name	Protection against Type of fault	Applied procedure for tolerate the fault
AFTRC[15]	Reliability	1.Delete node depending on their reliability 2.Back word recovery with the help of check pointing
LLFT[19]	Crash-cost, trimming fault	Replication.
FTWS[20]	Dead line of work flow	Replication and resubmission of jobs
FTM[21]	Reliability, availability, on demand service	Replication users application and in the case of replica failure use algorithm like gossip based protocol.
CANDY[22]	Availability	1. It assembles the model components generated from IBD and STM according to allocation notation. 2. Then activity SNR is synchronized to system SRN by identifying the relationship between action in activity SNR and state transition in system SRN.
VEGA-WARDEN[23]	Usability, security, scaling	1. Two layer authentication and standard technical solution for the application.
FT-CLOUD[24]	Reliability, crash and value fault	1. Significant component is determined based on the ranking. 2. Optimal ft technique is determined.
MAGI-CUBE[25]	Performance, reliability, low storage cost	1. Source file is encoded in then splits to save as a cluster. 2. File recovery procedure is triggered is the original file is lost.

3. Metrics for Fault-tolerance in Cloud Computing

The existing fault-tolerance technique in cloud computing consider various parameter. The parameters are like there type of fault-tolerance (proactive, reactive and adaptive), performance, response-time, scalability, throughput, reliability, availability, usability, security and associated over-head. Table 2 summarized the Comparison among various models based on the metrics element.

Different parameters including fault-tolerance type (proactive, reactive, and adaptive), performance, time response, scalability, through put, reliability, availability, usability, security, cost effectiveness and overhead associated are taken into account when dealing with the existing fault-tolerant methods in the Cloud Computation.

Proactive fault-tolerance: The Proactive fault-tolerance policy is to avoid recovery from fault, errors and failure by predicting them and proactively replace the suspected component means detect the problem before it actually come.

Reactive fault tolerance: Reactive fault-tolerance policies reduce the effort of failures when the failure effectively occurs. This technique provides robustness to a system.

Adaptive: All the procedure done automatically according to the situation.
Performance– This is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays.

Response Time - is the amount of time taken to respond by a particular algorithm. This parameter should be minimized.

Scalability–This is the ability of an algorithm to perform fault-tolerance for a system with any finite number of nodes. This metric should be improved.

Throughput–This is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.

Reliability: This aspect aims to give correct or acceptable result within a time bounded environment.

Availability: The probability that an item will operate satisfactorily at a given point with in time used under stated conditions. Availability of a system is typically measured as a factor of its reliability as reliability increases, so does availability.

Usability: The extent to which a product can be used by a user to achieve goals with effectiveness, efficiency, and satisfaction.

Overhead Associated: determines the amount of overhead involved while implementing a fault-tolerance algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should beminimized so that a fault-tolerance technique can work efficiently.

Cost effectiveness: Here the cost is only defined as a monitorial cost.
Table 2 summarizes the comparison among various models based on the metrics element.

Table 2. Comparison of Various Models based on the Metrics Element [26] (y=yes, n=no, h=high, l=low, a=average)

Model number	AFTRC	LLFT	FTWS	FTM	CANDY	VEGA-WARDEN	FT-CLOUD	MAGI-CUBE
Proactive	y	n	n	n	n	y	y	n
Reactive	n	y	y	y	y	y	n	y
Adaptive	y	n	n	n	y	n	y	y
Performance	h	h	a	a	a	h	h	h
Response time	a	a	a	a	a	h	a	a

Scalability	h	h	l	l	h	h	h	h
Through put	h	a	l	a	h	a	a	h
Reliability	h	h	a	a	h	h	h	h
Availability	h	h	a	h	h	h	a	a
Usability	h	a	a	a	a	h	h	h
Overhead Associated	a	l	h	l	l	h	h	a
Cost effectiveness	a	l	h	l	l	l	h	h

4. Conclusion and Further Work

Fault-tolerance methods work when a fault enters the boundary of a system. Therefore, in theory, fault-tolerance methods are used to predict the fault and perform an appropriate action, before the faults actually occur. This article discusses the classification of faults and the need to cover the fault-tolerance with various implementation techniques. Different fault-tolerance models are presented, and compared in terms of fault tolerance in the cloud. In the present scenario, there are tolerance error models that introduce different fault-tolerance mechanisms to improve the system. However, there are still challenges that need to be considered for any framework or model. There are weaknesses that cannot complete all aspects of the faults. Thus, it is possible to overcome the weaknesses of all previous models. Therefore, this article tries to create a compact model that covers the most aspects of fault-tolerance.

References

[1] M. A. Vouk, "Cloud Computing – Issues, Research and Implementations", Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA, Journal of Computing and Information Technology - CIT 16, vol. 4, **(2008)**, pp. 235–246.
[2] B. S. Taheri, M. G. Arani and M. Maeen, "ACCFLA: Access Control in Cloud Federation using Learning Automata", International Journal of Computer Applications, vol. 107, no. 6, **(2014)**, pp. 30-40.
[3] M. Fallah and M. G. Arani, "ASTAW: Auto-Scaling Threshold-based Approach for Web Application in Cloud Computing Environment", International Journal of u- and e- Service, Science and Technology (IJUNESST), vol. 8, no. 3, **(2015)**, pp. 221-230.
[4] A. Fereydooni, M. G. Arani and M. Shamsi, "EDLT: An Extended DLT to Enhance Load Balancing in Cloud Computing", International Journal of Computer Applications, vo. 108, no. 7, **(2014)**, pp. 6-11.
[5] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper 1st Edition, **(2009)**.
[6] M. Fallah, M. G. Arani and M. Maeen, "NASLA: Novel Auto Scaling Approach based on Learning Automata for Web Application in Cloud Computing Environment", International Journal of Computer Applications, vol. 113, no. 2, **(2015)**, pp. 18-23.
[7] A. Bala and I. Chana, "Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing", IJCSI International Journal of Computer Science Issues, vol. 9, no. 1, **(2012)**, pp. 1694-0814.
[8] B. Lussier, A. Lampe, R. Chatila, J. Guiochet, F. Ingrand, M. O. Killijian and D. Powell, "Fault Tolerance in Autonomous Systems: How and How much?", LAAS-CNRS 7 Avenue du Colonel Roche, F-31077 Toulouse Cedex 04, France.
[9] P. Latchoumy and S. A. Khader, "Survey on fault tolerance in grid computing", IJCSI International Journal of Computer Science Issues, vol. 2, no. 4, **(2011)**.
[10] M. C. Chan, J. R. Jiang and S. T. Huang, "Fault tolerant and secure networked storage", 7thInternational Conference on Digital Information Management, ICDIM, **(2012)**.
[11] F. Shieh, M. G. Arani and M. Shamsi, "De-duplication Approaches in Cloud Computing Environment: A Survey", International Journal of Computer Applications, vol. 120, no. 13, **(2015)**, pp. 6-10.
[12] J. Evans, "Fault Tolerance in Hadoop for Work Migration", **(2011)**.
[13] F. Hupfeld, T. Cortes and B. Kolbeck, "The XtreemFS architecture - A case for object-based file systems

in Grids", Concurrency Computation Practice and Experience, vol. 20, no. 17, **(2008)**, pp. 2049- 2060.
[14] F. Shieh, M. G. Arani and M. Shamsi, "An Extended Approach for Efficient Data Storage in Cloud Computing Environment", IJCNIS, vol. 7, no. 8, **(2015)**, pp. 30-38.
[15] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper 1st Edition, **(2009)**.
[16] J. C. Laprie, "Dependable computing and fault tolerance: concepts and terminology", LAAS-CNRS 7 Avenue du Colonel Roche, 31400 Toulouse, France.
[17] G. M. Nayeem and M. J. Alam, "Analysis of Different Software Fault Tolerance Techniques", **(2006)**.
[18] G. Vallee, K. Charoenpornwattana, C. Engelmann, A. Tikotekar and S. L. Scott, "A Framework for Proactive Fault Tolerance".
[19] W. B. Zhao, P. M. Melliar and L. E. Mose, "Fault Tolerance Middleware for Cloud Computing", IEEE 3rd International Conference on Cloud Computing, **(2010)**.
[20] S. K. Jayadivya, J. S. Nirmala and M. S. Bhanus, "Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing", International Journal on Computer Science and Engineering (IJCSE).
[21] R. Jhawar, V. Piuri and M. Santambrogio, "A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing", IEEE.
[22] F. Machida, E. Andrade, D. S. Kim and K. S. Trivedi, “Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML”, 30th IEEE International Symposium on Reliable Distributed Systems, **(2011)**.
[23] J. Lin, X. Y. Lu, L. Yu, Y. Q. Zou and L. Zha, "Vega Warden: A Uniform User Management System for Cloud Applications ", Fifth IEEE International Conference on Networking, Architecture, and Storage, **(2010)**.
[24] Z. B. Zheng, T. C. Zhou, M. R. Lyu and I. King, "FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications", IEEE 21st International Symposium on Software Reliability Engineering, **(2010)**.
[25] Q. Q. Feng, J. Z. Han, Y. Gao and D. Meng, "Magi-cube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing", IEEE Seventh International Conference on Networking, Architecture, and Storage, **(2012)**.
[26] P. K. Patra, H. Singh and G. Singh, "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing", International Journal of Computer Applications, pp. 0975 –8887.

Authors

Seyed Mansour Hosseini, he received the B.S.C degree in Software Engineering from University Azad Kashan, Iran in 2010, and M.S.C degree from Azad University of Mahallat, Iran in 2014, respectively. Her research interests include Cloud Computing, Distributed Systems, Cloud Storage Technology and Software Development.

Mostafa Ghobaei Arani, he received the B.S.C degree in Software Engineering from IAU Kashan, Iran in 2009, and M.S.C degree from Azad University of Tehran, Iran in 2011, respectively. He’s currently a PhD Candidate in Islamic Azad University, Science and Research Branch, Tehran, Iran. His research interests include Grid Computing, Cloud Computing, Pervasive Computing, Distributed Systems and Software Development.