International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-9, December 2015
ISSN: 2395-3470
www.ijseas.com

Survey on Fault Tolerance Techniques in Cloud Computing Environment

V.M.Sivagami
Associate Professor
Department of Information Technology
Sri Venkateswara College of Engineering ,Pennalur Sriperumbdur,India,Pin-602117.
vmsiva@svce.ac.in
Dr.K.S.EaswaraKumar
Professor
Department of Computer Science Engineering,
Anna University,Chennai-25
India
easwaracs@annauniv.edu

***Abstract*--Cloud computing offer IT services to the users worldwide on the basis of pay-as-you-go model. Cloud computing refer to network based services which appear to be provided by real server hardware, which in fact are served up by virtual hardware, simulated by software running on one or more real machines. These networked based services and systems are more or less prone to failures. In order to achieve robustness and dependability in cloud computing infrastructure, the failure should be assessed and handled effectively. This paper aims to provide a better understanding of various types of fault tolerance and fault tolerance techniques used in the cloud computing environment.**

***Keywords*- Cloud computing, Fault, errors, failures, fault tolerance techniques.**

I. INTRODUCTION

Cloud computing is a way of computing where service is provided across the internet using models and levels of abstraction [1]. Cloud computing is the combination of grid computing and utility computing [2].According to NIST "Cloud computing is a model for enabling ubiquitous, 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 "[3]. Many research issues are fully addressed in cloud such as Fault tolerance, security, etc. The flexibility of cloud computing is a function of allocating resource on demand. It has the capacity to yoke the internet and wide area network to use the resources that are available remotely there by to provide cost efficient solution on pay per use basis[4][5].Fault tolerance is an important key issue in cloud computing and it is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. The fault tolerance technique enables robustness and dependability in cloud environment. The major benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics etc.[5] The key motivation of the survey is find out the various existing fault tolerance techniques and models in cloud computing is to support researcher to contribute in developing more efficient algorithm. This paper is planned to deliberates about various aspect of faults and the need of fault tolerance in cloud computing.

II. FAULT–TOLERANCE IN CLOUD COMPUTING

Fault tolerance refers to correct and continuous operation even in the presence of faulty components. It is the art and science of building computing systems that continue to operate satisfactorily in the presence of faults. A fault tolerant system may be able to tolerate one or more fault types including- transient, intermittent or permanent hardware faults, software and design errors, operator errors, or externally induced upsets or physical damage[6].In real time cloud applications, processing on computing nodes is done remotely which has a high probability of occurrence of errors. These events increase the need for fault tolerance techniques to achieve reliability for the real time computing on cloud infrastructure**.**

The relationship between faults, error and failure is described in the below figure. Each is depended on each other. Occurrences of fault either in the component (Hardware) or in designing methodologies (Software) prone to error leads to the failure.[7]

Figure 1: Generation path of failure

- A system is said to be fail when it will not fulfill the requirements
- An error is the part of the system state that may lead to failure
- The cause of an error is a fault.

2.1TYPES OF FAULTS

The faults can be classified based on several factors.

- **Network fault:** A Fault that occurs in a network due to network partition, Congestion, packet loss, packet corruption, destination failure, link failure., etc.
- **Physical faults:** This Fault can occur in hardware like fault in CPUs, Fault in memory, Fault in storage, Power failure etc.
- **Media faults:** Fault that occurs due to media head crashes in storage media.
- **Process faults:** A fault which occurs due to shortage of resource, software bugs, inefficient processing capabilities etc.
- **Service expiry fault:** The service time of a resource may expire while application is using it.

A fault can be classified on the basis of computing resources and time. Failures that occur during computation on system resources can be classified as timing failure, omission failure, response failure and crash failure.

Failures may be permanent, Intermittent and transient.

- **Permanent failure:** These failures occur by accidental power breakdowns, natural disasters., etc. These failures can cause major disruptions and some part of the system may not be functioning as desired.
- **Intermittent failure:** These failures happen occasionally. These failures happen while the system performs its operations. The damage caused due to these failures are very difficult to be identified.
- **Transient failure:** These failures are caused by some inherent fault in the system and they can be easily resolved. These failure can be corrected by Retrying /Roll back the system to previous state. These failures are common in the computer system.

2.2 TYPES OF FAULT TOLERANCE

The fault tolerance is majorly classified in to two types. They are hardware and software fault tolerance.

Hardware fault tolerance:

Most of the computer systems are directed automatically to recover from failure that occur due to hardware components. Each and every component is backed up with protective redundancy, so that if one component fails the other can be alternated to perform the functions with any failure. Example: Mirroring technique which recovers the failure of storage media. The general hardware fault tolerance approaches are fault masking and dynamic recovery.

Fault Masking:

Fault masking is a structural redundancy technique that completely masks faults within a set of redundant components. A number of identical components execute the same functions, and their outputs are voted to remove errors created by a faulty module. Triple modular redundancy (TMR) is a commonly used form of fault masking in which the Component is triplicated and voted. The voting process for selecting the redundant component can also be triplicated so that individual voter failures can also be corrected by the voting process. A TMR approach fails whenever two components in a redundant triplet failed so that the vote is no longer valid. Hybrid redundancy is an extension of TMR in which the triplicated Components are backed up with additional components, which are used to replace faulty Components which allow more faults to be tolerated. Voted systems require more than three times as much hardware as non redundant systems, but they have the advantage that computations can continue without interruption when a fault occurs, allowing existing operating system to be used[8].

Dynamic recovery:

Dynamic recovery technique is used only when one copy of the task or computation is made to run at a time. This technique perform self-repair. As in fault masking technique, additional spare components are used to perform back up operation.(proactive redundancy). In this dynamic recovery technique a special recovery mechanism are required to detect faults in module. Instead of switching to redundant component the fault in the module or the component is analyzed and recovered from them using the actions like rollback, initialization, retry, restart. These actions perform the restore operation and continue computations. Dynamic recovery is more efficient than TMR approach, because it based on resource constrained.

Software fault tolerance:

The faults in the software can be tolerated by using static and dynamic approaches similar to the hardware fault tolerance approaches. N-version programming uses static redundancy in the form of independent written programs that perform the same operation. In an N-version programming, each module is made with up to N different implementations. Each programmer accomplishes the same task, but hopefully in a different way. Each version then submits its answer to voter or decider which determines the correct answer and returns that as the result of the module. An alternative dynamic approach is recovery blocks. In this technique programs are partitioned in to blocks and acceptance test are executed after each block. If the acceptance test fails, a redundant code block is executed. Design diversity approach combines both of hardware and software fault tolerance by implementing a fault tolerant computer system using different hardware and software in redundant channels[7].The main goal of design diversity technique is to tolerate both hardware and software faults but it is too expensive.

III FAULT TOLERANCE TECHNIQUES

Based on fault tolerance, different fault tolerance techniques and strategies are classified as follows:

Proactive fault tolerance: The principle of Proactive fault tolerance polices is to predict the fault and avoid recovery from fault, errors and failures proactively replace the suspected component (i.e) it detects the problem before it actually comes. It prevents compute node failures from impacting running parallel applications by preemptively migrating parts of an application (task, process, or virtual machine) away from nodes that are about to fail. Some of the techniques based on these policies are Preemptive migration, Software Rejuvenation and using self-healing [8].

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.

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

Reactive Fault Tolerance: Reactive fault tolerance policies reduce the effect of failures on application execution when the failure effectively occurs. It is also called as on-demand fault tolerance. There are various techniques based on these policies like Checkpoint/Restart, Replay-retry, task resubmission, recue workflow, user defined exception handling, retry, S-Guard, job migration etc. [9][10][11][12][13]

Check pointing: It is an efficient task level fault tolerance technique for long running and big applications .In this method 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 from the recently checked pointed state.

Job Migration :Some time it happens that due to some reason a job cannot 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: Replication means copy. Various tasks are replicated and they run on different resources, for the successful execution and for getting the desired result. Using tools like HA-Proxy, Hadoop and AmazonEc2 replication can be implemented.

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

S-Guard: It is less turbulent to normal stream processing. S-Guard is based on rollback recovery. S-Guard can be implemented in HADOOP and 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 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.

Adaptive fault tolerance: The fault-tolerance of an application need to be changed depending on range of control inputs and the current position in its state space. Adaptive fault tolerance automatically invokes the procedures to control the situation and they assure adequate reliability of critical modules under any resources and temporal constraints by allocating as much redundancy resources and modules.

IV FAULT TOLERANCE MODELS

AFTRC - Adaptive fault tolerance Real-time computing model is for real time application which has a high processing capabilities in cloud computing environment. In this model, the system tolerates the fault proactively and makes the diction on the basis of reliability of the processing nodes.[14]

LLFT : Low latency fault tolerance model is a middleware for providing fault tolerance in distributed applications that are deployed in the cloud environment. This middleware provide fault tolerance by replication .The application uses semi-active replication or semi-passive replication process to protect against various types of faults.[15]

FTM : Fault tolerance model is proposed to overcome the limitation of existing methodologies of on-demand service. This model ensures reliability and resilience by using innovative methodology through which the user can specify and apply the desired level of fault tolerance without requiring any information about its implementation.FTM can be viewed as an assembly of several web services components, each with a specific functionality[16].

FTWS : Fault tolerance workflow scheduling model contains a fault tolerant work flow scheduling algorithm to provide fault tolerance by using replication and resubmission of tasks based on the priority of the tasks in a heuristic matrix.
This model is based on workflow - a set of tasks processed in some order based on data and control dependency. The scheduling of workflow also considers the task failure in the cloud environment. FTWS replicates and schedule the tasks to meet their deadlines.[17]

Candy - Candy is a component based availability modeling frame work which constructs a comprehensive availability model semi automatically from system specification described 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 [18].

Vega-warden is a uniform user management system which provides a global user space for different virtual infrastructure and application services in cloud computing environment. This model is extremely used for virtual cluster based cloud computing environment to overcome the usability and security problems arises from sharing of infrastructure [19].

FT-Cloud is a component ranking based frame work and its architecture is used for building cloud application. FT-Cloud employs the component invocation structure and frequency to identify the component. There is an algorithm to automatically determine fault tolerance stately [20].

Magi-Cube: is a highly reliable and low redundancy storage architecture for cloud computing. They build the system on 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 which provides high reliability and performance at low cost.[21]

V TOOLS USED FOR IMPLEMENTING FAULT TOLERANCE

Fault tolerance techniques can be implemented by using various tools. Some of the tools are listed below.

- HAProxy is used for handling server failover in the cloud [22].
- SHelp [23] is a lightweight runtime system that can survive software failures in the framework of virtual machines. It can be also used to implement checkpoint in cloud environment.
- ASSURE [24] uses rescue points for handling programmer anticipated failures in cloud environment.
- Hadoop [25] is used for data intensive applications and can also be used to implement fault tolerance techniques in cloud environment.
- Amazon Elastic Compute Cloud (EC2) [26] introduces a virtual computing environment to run Linux-based applications for fault tolerance.

Table 1: Comparison of various fault tolerance models.

Model Name	Protection against the type of fault	Used Procedures
AFTRC	Reliability	Check pointing with back word recovery Deletion of nodes based on reliability
LLFT	Crash cost and trimming fault	Reliability
FTM	Reliability, Availability and on demand service	Replication of user application and uses Gossiping protocol incase of redundancy failure.
FTWS	Task deadline	Replication and resubmission of jobs
CANDY	Availability	It assembles the model components generated from IBD and STM according to allocation notation. 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	Usability, Scalability and security	Two layer authentication and technical solution for the application
FT-CLOUD	Reliability, Crash and value fault	Significant component is determined based on the ranking and Optimal FT technique is determined.
MAGI-CUBE	Performance, reliability and low storage cost	Source file is encoded and it splits to save as a cluster. File recovery procedure is triggered if the original file is lost.

Table 2: Comparison of various fault tolerance tools:

Fault Tolerance Techniques	Policies	System	Programming Framework	Environment	Fault detected	Application on type
Self Healing, job migration, Replication	Reactive/ Proactive	HAProxy	Java	Virtual machines	Process/node failures	Load balancing fault tolerance
Check pointing	Reactive	SHelp	SQL,JAVA	Virtual machines	Application failures	Fault tolerance

Check pointing, Retry, Self Healing	Reactive/ Proactive	Assure	JAVA	Virtual machines	Host/Network failures	Fault tolerance
Job Migration, Replication, S-Guard	Reactive/ Proactive	Hadoop	Java,HTML,CSS	Virtual machines	Application/node failures	Data intensive
Replication, S-Guard, Task Resubmission	Reactive/ Proactive	Amazon EC2	Amazon Machine Image, Amazon Map	Virtual machines	Application/node failures	Load balancing and fault tolerance

VI CONCLUSION

Fault tolerance is about tolerating, avoiding or improving faults and errors remained in the system after its development and implementation. This paper discussed the types of faults and various types of fault tolerance techniques and comparison of various fault tolerance techniques. This paper also discussed about the various tools used for fault tolerance and their comparison. In future, the identification of failures in virtual machines and changes that happens in cloud environment due to failures are identified and algorithms may be proposed to eliminate the same.

VII REFERENCES

[1] L. Arockiam, S. Monikandan& G. Parthasarathy," Cloud Computing: A Survey, International Journal of Internet Computing (IJIC), and ISSN No: 2231 – 6965, Volume-1, Issue-2, 2011.

[2] Qi Zhang, Lu Cheng,RaoufBoutaba," Cloud computing: state-of-the-art and research challenges, J Internet ServAppl (2010) 1: 7–18 DOI 10.1007/s13174-010-0007-6.

[3] Peter Mell, Timothy Grance, "NIST Definition of Cloud Computing", Sept 2011, National Institute of Standards and technology, Gaithersburg, MD 20899-8930.

[4] Sun Microsystems, Inc. "Introduction to Cloud Computing Architecture" White Paper 1st Edition, June 2009

[5] Mladen 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, 2008, 4, 235–246doi:10.2498 /cit.100139

[6] Ravi Jhawar, Vincenzo Piuri, Marco Santambrogio, " A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing"© 2012 IEEE, DOI 10.1109/SysCon.2012.6189503

[7] Amritpal Singh, Supriya Kinger, "An Efficient Fault Tolerance Mechanism Based on Moving Averages Algorithm" © 2013, IJARCSSE, ISSN: 2277 128X

[8] http://www.cs.ucla.edu/~rennels/article98.pdf

[9] Anju Bala, Inderveer Chana, "Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing", IJCSI international Journal of Computer Science Issues, Vol.9, Issue 1, No 1, January 2012

[10] Benjamin Lussier, Alexandre Lampe, Raja Chatila, Jérémie Guiochet, Félix Ingrand, Marc-Olivier Killijian, David Powell, "Fault Tolerance in Autonomous Systems: How and How Much?" LAAS-CNRS 7 Avenue du Colonel Roche, F-31077 Toulouse Cedex 04, France

[11] Jean-clande Laprie "Dependable computing and fault tolerance: concepts and terminology" LAAS-CNRS 7 Avenue du Colonel Roche, 31400 Toulouse, France

[12] Patel et al., International Journal of Advanced Research in Computer Science and Software Engineering 3(12), December - 2013, pp. 573-576

[13] Wenbing Zhao, P.M. Melliar and L.E. Mose" Fault Tolerance Middleware for Cloud Computing" 2010 IEEE 3rd International Conference on Cloud Computing.

[14] Sheheryar MalikandFabriceHuet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Service

[15] Wenbing Zhao, P.M. Melliar and L.E. Mose" Fault Tolerance Middleware for Cloud Computing" 2010 IEEE 3rd International Conference on Cloud Computing

[16] Ravi Jhawar, Vincenzo Piuri and Marco Santambrogio"A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing" IEEE

[17] Jayadivya S K, JayaNirmala S, Mary SairaBhanus"Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing" International Journal on Computer Science and Engineering (IJCSE)

[18] Fumio Machida, Ermeson Andrade, Dong SeongKim and Kishor S. Trivedi"Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML" 2011 30th IEEE International Symposium on Reliable Distributed Systems.

[19] Jianlin, Xiaoyi Lu, Lin Yu, YongqiangZou and Li Zha"Vega Warden: A Uniform User Management System for Cloud Applications "2010 Fifth IEEE International Conference on Networking, Architecture, and Storage.

[20] ZibinZheng, Tom Chao Zhou, Michel R. Lyu, and Irwin king "FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications "2010 IEEE 21st International Symposium on Software Reliability Engineering.

[21] QingqingFeng, Jizhong Han, Yun Gao, Dan Meng"Magicube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing" 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage.

[22] http://haproxy.1wt.eu/download/1.3/doc/configuration.txt.

[23] Gang Chen, Hai Jin, Deqing Zou, Bing Bing Zhou, Weizhong Qiang, Gang Hu, "SHelp: Automatic Selfhealing for Multiple Application Instances in a Virtual Machine Environment", IEEE International Conference on Cluster Computing, 2010

[24] S. Sidiroglou, O. Laadan, C. Perez, N. Viennot, J. Nieh, and A. D. Keromytis, "ASSURE: Automatic Software Self-healing Using REscue points", Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating System.

[25] S. Sidiroglou, O. Laadan, C. Perez, N. Viennot, J. Nieh, and A. D. Keromytis, "ASSURE: Automatic Software Self-healing Using REscue points", Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating System.

[26] AmazonElasticComputeCloud(EC2) http://www.amazon.com/ec2/