Fault Tolerance Techniques and Comparative Implementation in Cloud Computing

Prasenjit Kumar Patra
Lovely Professional University
Phagwara, Punjab

Harshpreet Singh
Lovely Professional University
Phagwara, Punjab

Gurpreet Singh
Lovely Professional University
Phagwara, Punjab

ABSTRACT

Cloud computing is the result of evolution of on demand service in computing paradigms of large scale distributed computing. It is the adoptable technology as it provides integration of software and resources which are dynamically scalable. These systems are more or less prone to failure. Fault tolerance assesses the ability of a system to respond gracefully to an unexpected hardware or software failure. In order to achieve robustness and dependability in cloud computing, failure should be assessed and handled effectively. This paper aims to provide a better understanding of fault tolerance techniques used for fault tolerance in cloud environments along with some existing model and further compare them on various parameters.

Keywords

Cloud Computing; Fault Tolerance; Reactive; Proactive; Dependability; Reliability

1. INTRODUCTION

"Cloud computing"-emerging as a new paradigm of large scale distributed computing embraces cyber infrastructure and builds upon on the concept in virtualization, grid computing, utility computing, networking, web services and software services to implement a service oriented architecture for reducing information technology overhead for the end-user for provide great flexibility and reduced total cost of ownership and all above on-demand services to a shared pool of computing resources. 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 [1][2].Due to the rapid exponential growth of cloud computing the need of fault tolerance in cloud is an key factor for concern.

Fault tolerance bear-on with all the inevitably techniques to enable robustness and dependability .The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics [3]. Robustness leads to the property to providing of a correct service in an adverse situation arising due to an uncertain system environment [4]. Dependability is related to some QOS aspects provided by the system, it includes the attributes like reliability and availability [5].

The motivation of the survey of existing fault tolerance techniques and models in cloud computing is to encourage researcher to contribute in developing more efficient algorithm. This paper is organized to discusses about various aspect of faults and the need of fault tolerance in cloud computing

2. FAULT TAXONOMY AND NEED OF FAULT TOLERANCE IN CLOUD COMPUTING

Fault tolerance aim to achieve robustness and dependability in any system. Based on fault tolerance policies and techniques we can classify this technique into 2 types: proactive and reactive. 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 policies reduce the effort of failures when the failure effectively occurs. These can be further classified into two sub-techniques error processing and fault treatment. Error processing aims at removing errors from the computational state. Fault treatment aims at preventing faults from being re-activated [4] [5].

Fault tolerance is carried out by error processing which have two constituent phases. The phases are "effective error processing" which aimed at bringing the effective error back to a latent state, if possible before occurrence of a failure and "latent error processing" aimed at ensuring that the error does not become effective again [6].

3. EXISTING FAULT TOLERANCE TECHNIQUES IN CLUD COMPUTING

Various fault tolerance techniques are currently prevalent in clouds [3] [4] [6] [7] [8]:-

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-Replication means copy. Various tasks are replicated and they are 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.

Self- Healing- A big task can divided into parts .This Multiplicationis 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 [4].

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 [4].

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 [6].

Reconfiguration: In this procedure we eliminate the faulty 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.

 Several models are implemented based on these types of techniques. Table 1 summarized the Comparison among various models based on protection against the type of fault, and procedure.

"**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 [9].

"**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 [10].

"**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 [11].

"**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 [12].

"**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 [13].

"**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 [14].

"**FT-Cloud**" is 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 [15].

"**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 [16].

Table 1: Comparison among various models based on protection against the type of fault, and procedure

Model no	Model name	Protection against Type of fault	Applied procedure for tolerate the fault
M1	AFTRC	Reliability	1.Delete node depending on their reliability 2.Back word recovery with the help of check pointing
M2	LLFT	Crash-cost, trimming fault	Replication.
M3	FTWS	Dead line of work flow	Replication and resubmission of jobs
M4	FTM	Reliability, availability, on demand service	Replication users application and in the case of replica failure use algorithm like gossip based protocol.
M5	CANDY	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.
M6	VEGA-WARDEN	Usability, security, scaling	1. Two layer authentication and standard technical solution for the application.
M7	FT-CLOUD	Reliability, crash and value fault	1. Significant component is determined based on the ranking. 2. Optimal ft technique is determined.
M8	MAGI-CUBE	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.

4. 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.

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 be

minimized so that a fault tolerance technique can work efficiently

Cost effectiveness: Here the cost is only defined as a monitorial cost.

Table 2: Comparison among various models based on the metrics element

Model number	M1	M2	M3	M4	M5	M6	M7	M8
Proactive(y/n)	y	n	n	n	n	y	y	n
reactive(y/n)	n	y	y	y	y	y	n	y
adaptive(y/n)	y	n	n	n	y	n	y	y
Performance(h/l/a)	h	h	a	a	a	h	h	h
Response time(h/l/a)	a	a	a	a	a	h	a	a
Scalability(h/l/a)	h	h	l	l	h	h	h	h
Through put(h/l/a)	h	a	l	a	h	a	a	h
Reliability(h/l/a)	h	h	a	a	h	h	h	h
Availability(h/l/a)	h	h	a	h	h	h	a	a
Usability(h/l/a)	h	a	a	a	a	h	h	h
Overhead Associated(h/l/a)	a	l	h	l	l	h	h	a
Cost effectiveness(h/l/a)	a	l	h	l	l	l	h	h

(y=yes, n=no, h=high, l=low, a=average)

5. CONCLUSION

Fault tolerance methods come into play the moment a fault enters the system boundaries. So theoretically fault tolerance techniques are used to predict these failures and take an appropriate action before failures actually occur. This paper discusses about the fault taxonomy and need of fault tolerance covering with its various techniques for implementing fault tolerance. Various proposed models for fault tolerance are discussed and compared on the basis of Metrics for fault tolerance in cloud. In the present scenario, there are number of fault tolerance models which provide different fault tolerance mechanisms to enhance the system. But still there are number of challenges which need some concern for every frame work or model. There are some drawback no one of them can full fill the all aspects of faults. So there is a possibility to overcome the drawbacks of all previous models and try to make a compact model which will cover maximum fault tolerance aspect.

6. ACKNOWLEDGEMENT

7. REFERENCES

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

[2] 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.1001391

[3] AnjuBala, InderveerChana," Fault Tolerance-Challenges, Techniques and Implementation in Cloud

Computing" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012 ISSN (Online): 1694-0814 www.IJCSI.org

[4] Benjamin Lussier, Alexandre Lampe, Raja Chatila, JérémieGuiochet, 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 {firstname.lastname}@laas.fr

[5] P.LatchoumyandP.Sheik AbdulKhader,"Survey on fault tolerance in grid computing" IJCSI International Journal of Computer Science Issues, Vol 2, No 4, November 2011.

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

[7] GolamMoktaderNayeem , Mohammad Jahangir Alam," Analysis of Different Software Fault Tolerance Techniques", 2006.

[8] GeoffroyVallee, KulathepCharoenpornwattana, Christian Engelmann, AnandTikotekar, Stephen L. Scott," A Framework for Proactive Fault Tolerance".

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

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

[11] 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)

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

[13] 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.

[14] 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.

[15] 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.

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