Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing

Thakur Kapil Singh[1], Godavarthi Tarakarama RaviTeja[2], Padmavathi Srinivasa Pappala[3]

[1] Information Technology Department, Gitam University, Visakhapatanam-530045, Andhra Pradesh, India
[**] Information Technology Department, Gitam University, Visakhapatanam, Andhra Pradesh, India
[**] Information Technology Department, Gitam University, Visakhapatanam, Andhra Pradesh, India

Abstract- Fault tolerance is a major concern to guarantee availability and reliability of critical services as well as application execution. In order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. Fault tolerance techniques are used to predict these failures and take an appropriate action before failures actually occur. This paper discusses the existing fault tolerance techniques in cloud computing based on their policies, tools used and research challenges. Cloud virtualized system architecture has been proposed. In the proposed system autonomic fault tolerance has been implemented. The experimental results demonstrate that the proposed system can deal with various software faults for server applications in a cloud virtualized environment.

Index Terms- Cloud Computing; Virtual Machine; Fault Tolerance; Replication

I. INTRODUCTION

Cloud computing is a style of computing where service is provided across the Internet using different models and layers of abstraction [4]. It refers to the applications delivered as services [5] to the mass, ranging from the end-users hosting their personal documents on the Internet to enterprises outsourcing their entire IT infrastructure to external data centers. A simple example of cloud computing service is Yahoo email or Gmail etc.

Although cloud computing has been widely adopted by the industry, still there are many research issues to be fully addressed like fault tolerance, workflow scheduling, workflow management, security etc. Fault tolerance is one of the key issues amongst all. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. When a fault occurs, which are based on these policies like Checkpoint/Restart, Replay and Retry and so on. These techniques provide mechanisms to the software system to prevent system failure occurrence [18]. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics etc. This paper aims to provide a better understanding of fault tolerance challenges and identifies various tools and techniques used for fault tolerance. When multiple instances of an application are running on several virtual machines and one of the server goes down, there is a need to implement an autonomic fault tolerance technique that can handle these types of faults. To address this issue, cloud virtualized system architecture has been proposed and implemented using HAProxy. The proposed architecture also has been validated through experimental results.

The rest of the paper is organized as follows. Section 2 discusses fault tolerance techniques based on their policies. Section 3 presents challenges of implementing fault tolerance in cloud computing. Section 4 identifies the comparison between various tools used for implementing fault tolerance techniques with their comparison table. Section 5 presents proposed cloud virtualized architecture and implementation with experimental results. Section 6 finally concludes the paper.

II. BACKGROUND

There are various faults which can occur in cloud computing. Based on fault tolerance policies various fault tolerance techniques can be used that can either be task level or workflow level.

2.1 Reactive fault tolerance

Reactive fault tolerance policies reduce the effect of failures on application execution when the failure effectively occurs. There are various techniques Check pointing/ Restart - When a task fails, it is allowed to be restarted from the recently checked pointed state rather than from the beginning. It is an efficient task level fault tolerance technique for long running applications [2].

Replication-Various task replicas are run on different resources, for the execution to succeed till the entire replicated task is not crashed. It can be implemented using tools like HAProxy, Hadoop and AmazonEc2 etc.

Job Migration-During failure of any task, it can be migrated to another machine. This technique can be implemented by using HAProxy.

SGuard- It is less disruptive to normal stream processing and makes more resources available. SGuard is based on rollback recovery [18] and can be implemented in HADOOP, Amazon EC2.

Retry-It is the simplest task level technique that retries the failed task on the same cloud resource [20].

Task Resubmission-It is the most widely used fault tolerance technique in current scientific workflow systems [25]. Whenever a failed task is detected, it is resubmitted either to the same or to a different resource at runtime.

User defined exception handling-In this user specifies the particular treatment of a task failure for workflows.

Rescue workflow-This technique [20] allows the workflow to continue even if the task fails until it becomes impossible to move forward without catering the failed task.

2.2 Proactive Fault Tolerance

The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and failures by predicting them and proactively replace the suspected components other working components. Some of the techniques which are based on these policies are Preemptive migration, Software Rejuvenation etc.

Software Rejuvenation-It is a technique that designs the system for periodic reboots. It restarts the system with clean state [5].

Hadoop [7] is used for data intensive applications but can also be used to implement fault tolerance techniques in cloud enviorment.Amazon Elastic Compute Cloud (EC2) [8] provides a virtual computing environment to run Linux-based applications for fault tolerance.

Proactive Fault Tolerance using Self-Healing- When multiple instances of an application are running on multiple virtual machines, it automatically handles failure of application instances.

Proactive Fault Tolerance using Preemptive Migration-Preemptive Migration relies on a feedback-loop control mechanism where application is constantly monitored and analyzed.

III. Challenges of Implementing Fault Tolerance in Cloud Computing

Providing fault tolerance requires careful consideration and analysis because of their complexity, inter-dependability and the following reasons.

There is a need to implement autonomic fault tolerance technique for multiple instances of an application running on several virtual machines [12].

Different technologies from competing vendors of cloud infrastructure need to be integrated for establishing a reliable system [15].

The new approach needs to be developed that integrate these fault tolerance techniques with existing workflow scheduling algorithms [14].

A benchmark based method can be developed in cloud environment for evaluating the performances of fault tolerance component in comparison with similar ones [21].

To ensure high reliability and availability multiple clouds computing providers with independent software stacks should be used [22] [23].

Autonomic fault tolerance must react to synchronization among various clouds [15].

IV. Tools Used For Implementing Fault Tolerance

Fault tolerance challenges and techniques have been implemented using various tools. Table 1 compares these tools based on their programming framework, environment and application type along with different fault tolerance techniques. HA Proxy is used for server failover in the cloud [13]. *S*Help [12] is a lightweight runtime system that can survive software failures in the framework of virtual machines. It may also work in cloud environment for implementing check pointing. ASSURE [9] introduces rescue points for handling programmer anticipated failures.

Table 1: Tools Used To Implement Existing Fault Tolerance Techniques

Fault Tolerance Techniques	***Policies***	***System***	***Programming Framework***	***Environment***	***Fault Detected***	***Application Type***
Self Healing, Job Migration, Replication	Reactive/ Proactive	HAProxy[13]	Java	Virtual Machine	Process/node failures	Load balancing Fault Tolerance
Check pointing	Reactive	SHelp[12]	SQL,JAVA	Virtual Machine	Application Failure	Fault tolerance
Check pointing, Retry, Self Healing	Reactive/ Proactive	Assure[9]	JAVA	Virtual Machine	Host, Network Failure	Fault tolerance
Job Migration,Replication,Sguard,Resc	Reactive/ Proactive	Hadoop[7]	Java,HTML,CSS	Cloud Environment	Application/node failures	Data intensive
Replication, Sguard,Task Resubmission	Reactive/ Proactive	AmazonEC2[8]	Amazon Machine Image, Amazon Map	Cloud Environment	Application/node failures	Load balancing, fault

						tolerance

V. Proposed Cloud Virtualized System Architecture And Implementation

5.1 Cloud Virtualized System Architecture

A few techniques currently exist for autonomic fault tolerance in cloud enviorment.Shelp can survive the software faults for server applications running in virtual machine environment [12].There is a need to implement autonomic fault tolerance in cloud enviorment.If any one of the servers breaks down, system should automatically redirect user requests to the backup server. So, the cloud virtualized system architecture has been proposed and implemented using HAProxy.The application availability and reliability can be maintained by using the proposed cloud virtualized system architecture as shown in Figure 1. The server virtualized system consists of VMs (server 1 and server 2) on which an Ubuntu 10.04 OS and database application are running. Server 2 is a backup sever in case of failure. HAProxy is configured on the third virtual machine to be used for fault tolerance. The availability of the servers is continuously monitored by HAProxy statistics tool on a fault tolerant server. HAProxy is running on web server to handle requests from web. When one of the servers goes down unexpectedly, connection will automatically be redirected to the other server.

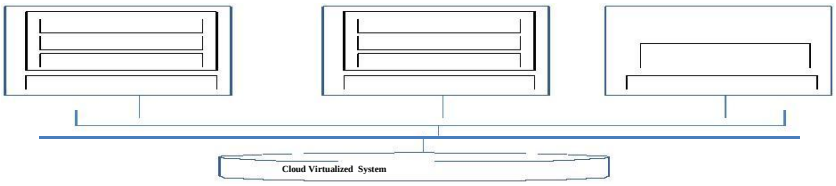

Figure 1: Cloud Virtualized System Architecture

5.2 Implementation and Experimental Results

Fault tolerant system has been implemented using HAProxy and MySQL. HAProxy is used to handle server failures in fault tolerant cloud environment. It provides a web interface for statistics known as HAProxy statistics. Implementation includes two virtual machines as web servers, server 1 and server 2 hosting Apache Tomcat 6.0.32. HAProxy software version 1.3.15.2 is configured on third virtual machine. A simple database application written in Java is installed on the web servers. Xampp for Linux Windows XP (SP2) is used to install MySQL. Data in MySQL is replicated using replication technique for local backup. Replication enables data from one MySQL database server (the master) to be replicated to one or more MySQL database servers. Application can be accessed on any of the web server. Data consistency is also maintained through MySQL replication. The experimental results show that HAProxy can assist server applications to recover from server failures in just a few milliseconds with minimum performance overhead.

Case 1: Figure 2 shows the stats when both the servers are working and green line indicates that the servers are up.

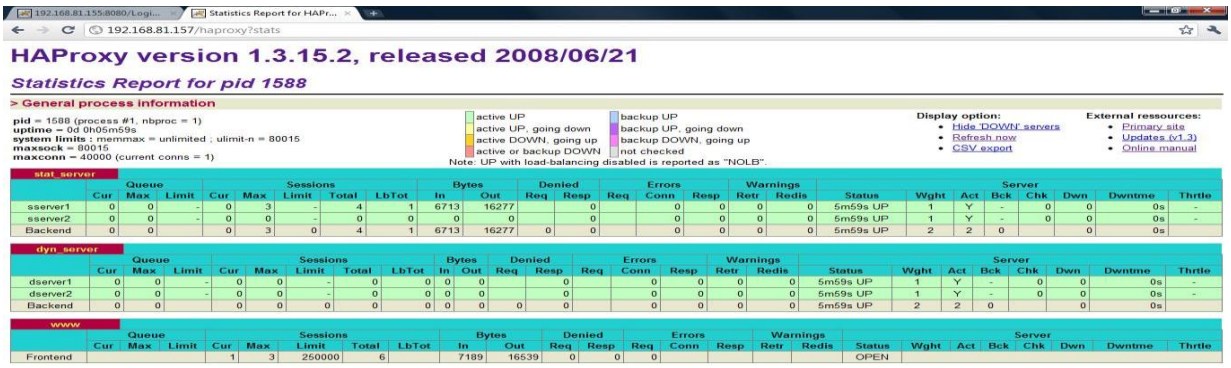

Figure 2: HAProxy Statistics

Case 2: Figure 3 shows the stats when server 1 goes down and server 2 is still up. Red line in this figure indicates that the server is down.

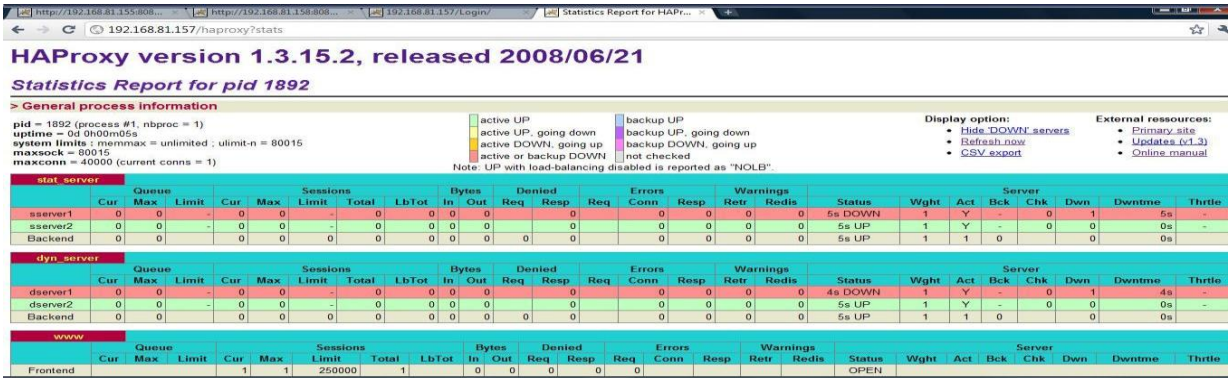

Figure3: Server 1 is down

Case 3: Figure 4 shows the stats when server 2 goes down and server 1 is still up.

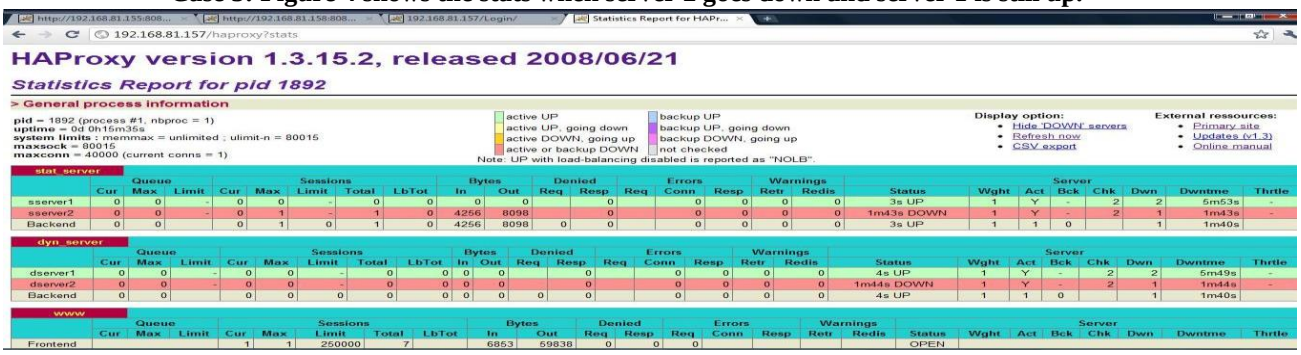

Figure 4: Server 2 is down

Case 4: Replication enables data from one MySQL database server to be replicated to one or more MySQL database servers (the slaves). Figure 5 shows the database table of server 1 using SQLyog and the connection established with server 2 slave.

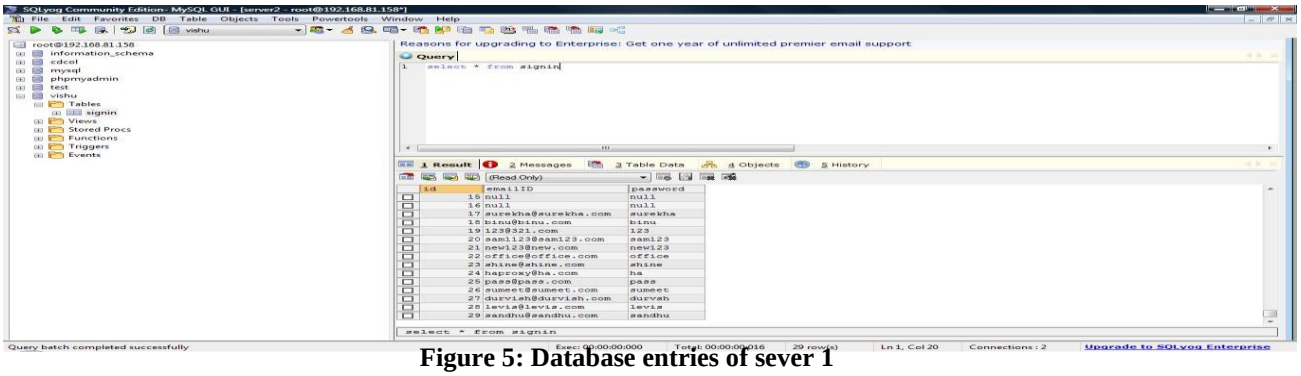

Figure 5: Database entries of sever 1

Case 5: Figure 6 shows the replicated database table of server 2.When server 1 fails data is replicated to server 2.

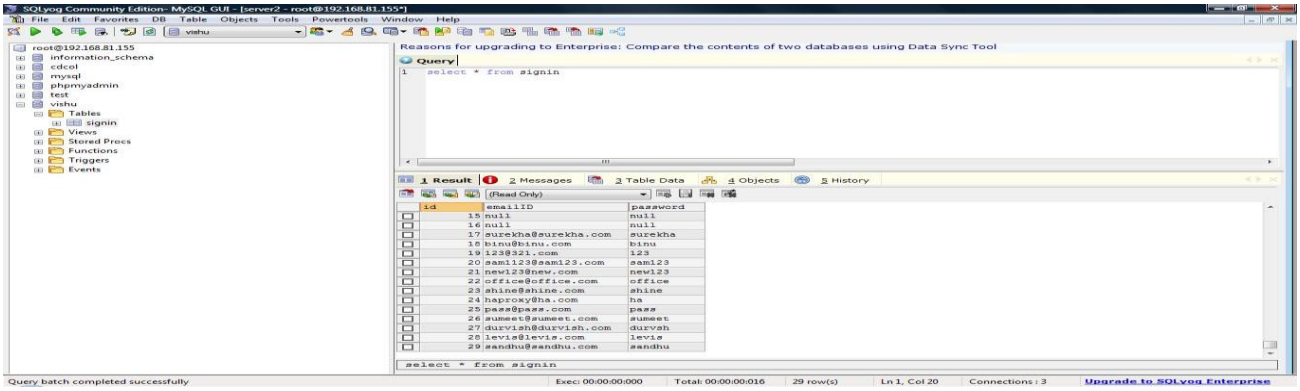

Figure 6: Replicated Database entries of sever 2

VI. CONCLUSION

Fault tolerance is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. This paper discussed the fault tolerance techniques covering its research challenges, tools used for implementing fault tolerance techniques in cloud computing. Cloud virtualized system architecture is also proposed based on HAProxy. Autonomic fault tolerance is implemented dealing with various software faults for server applications in a cloud virtualized environment. When one of the servers goes down unexpectedly, connection will automatically be redirected to the other server. Data replication technique is implemented on virtual machine environment. The experimental results are obtained, that validate the system fault tolerance.

REFERENCES

[1] Antonina Litvinova, Christian Engelmann and Stephen L. Scott," A Proactive Fault Tolerance Framework for High Performance Computing", 2009.

[2] Golam Moktader Nayeem, Mohammad Jahangir Alam," Analysis of Different Software Fault Tolerance Techniques", 2006.

[3] Steven Y. Ko, Imranul Hoque, Brian Cho and Indranil Gupta, "On Availability of Intermediate Data in Cloud Computations", 2010.

[4] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: towards a cloud definition," SIGCOMM Computer Communication Review,vol. 39, pp. 50–55, December 2008.

[5] M.Armbrust, A.Fox, R. Griffit,et al., "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[6] R.Buyya, S.Pandey and C.Vecchiola, "Cloudbus toolkit for market-oriented cloud computing", In Proceeding of the 1st International Conference on Cloud Computing (CloudCom2009), Beijing, China, December 2009.

[7] HadoopMapReduceTutorial.http://hadoop.apache.org/co re/docs/current/mapred tutorial.html.

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

[9] 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 Systems (ASPLOS'09), ACM Press, March 7-11, 2009, Washington, DC, USA, pp.37-48.

[10] B. Buck and J. K. Hollingsworth, "An API For Runtime Code Patching", International Journal of High Performance Computing Applications, Vol.14, No.4, November 2000, pp.317-329. pp.361-376.

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

[12]]http://haproxy.1wt.eu/download/1.3/doc/configuration.t xt.

[13] Yang Zhang1, Anirban Mandal2, Charles Koelbel1 and Keith Cooper," Combined Fault Tolerance and Scheduling Techniques for Workflow Applications on Computational Grids "in 9th IEEE/ACM international symposium on clustering and grid, 2010.

[14] Imad M. Abbadi, "Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure", 2010.

[15] Manish Pokharel and Jong Sou Park, "Increasing System Fault Tolerance with Software Rejuvenation in E-government System", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.

[16] Zaipeng Xie, Hongyu Sun and Kewal Saluja, "A Survey of Software Fault Tolerance Techniques".

[17] Geoffroy Vallee, Kulathep Charoenpornwattana, Christian Engelmann, Anand Tikotekar, Stephen L. Scott," A Framework for Proactive Fault Tolerance".

[18] Anglano C, Canonico M," Fault-tolerant scheduling for bag-of-tasks grid applications", In: Advances in grid computing—EGC 2005. Lecture notes in computer science, vol 3470/2005. Springer,Berlin/Heidelberg. ISSN: 0302-9743 Print. doi:10.1007/b137919, ISBN: 978-3-540-26918-2,pp 630–639

[19] Elvin Sindrilaru,,Alexandru Costan,, Valentin Cristea," Fault Tolerance and Recovery in Grid Workflow Management Systems", 2010 International Conference on Complex, Intelligent and Software Intensive Systems.

[20] S. Hwang, C. Kesselman, "Grid Workflow: A Flexible Failure Handling Framework for the Grid",12th IEEE international Symposium on Zigh Performance Distributed Computing (HPDC'03), Seattle, Washington,USA., IEEE CS Press, Los Alamitos, CA, USA, June 22 - 24, 2003.

[21] Michael Armbrust, Armando Fox,Rean Griffith, " Above the Clouds: A Berkeley View of Cloud Computing", Electrical Engineering and Computer Sciences University of California at Berkeley, 2009.

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

[23] Kassian Plankensteiner, Radu Prodan, Thomas Fahringer,"A New Fault Tolerance Heuristic for Scientific Workflows in Highly Distributed Environments based on Resubmission Impact", Fifth IEEE International Conference on e-Science,Austria,2009

AUTHORS

First Author – Thakur Kapil Singh, Information Technology Department, Gitam University, Visakhapatanam-530045, Andhra Pradesh, India

Second Author – Godavarthi Tarakarama RaviTeja, Information Technology Department, Gitam University, Visakhapatanam, Andhra Pradesh, India

Third Author – Padmavathi Srinivasa Pappala, Information Technology Department, Gitam University, Visakhapatanam, Andhra Pradesh, India