A Study on Fault Tolerance in Cloud Data Centers to Improve Performance Efficiency

V. Haripriya
Asst.Professor,Dept.of CS/IT
Jain University,Bangalore

Zhang Xvyi
Jain University,Bangalore

Dr. Suchithra R
Head,Dept.of MS(IT)
Jain University,Bangalore

Abstract:- **Cloud Computing has gained popularity in the recent years. Node failure when applications run in servers are a major concern to be addressed. This is otherwise called as Fault Tolerance.Reliability and availability of nodes during execution of critical service applications is a major concern which may otherwise affect the quality of service provided by the cloud service providers. In order to reduce the impact of failure when an application runs on the cloud, there should be mechanism to anticipate the failures so that failures can be proactively addressed. This paper address the various fault tolerance techniques used at deal with various software faults in a virtualized cloud environment.**

Keywords:- Virtualization, Fault Tolerance; virtual machine Replication

INTRODUCTION:

Companies outsource their IT services to third party providers due to the high cost of maintaining the internal infrastructure. This trend led to the emergence of the so-called cloud computing approach.These providers are called as Cloud Service Providers. One of the greatest advantages of cloud computing is to allow customers to pay only for the amount of resources used by them. The cloud provider is responsible for the administration of the cloud resources that includes hardware and virtual machines(VM)) and services. It is the responsibility of the provider to manage the accommodation of the capacity of the cloud. Cloud customers make use of the resources provided by the cloud to deploy and execute their applications.

When real time applications of customers run in cloud infrastructure the chance of node failure is quite high. As most of the application which run on cloud are safety critical systems. In general, real-time system is one that should process information and create a response within a scheduled time else may risk severe consequences including failure [1]]. So the reliability depends not only on the logical result, but also the time of delivery [2]. Failure to respond when a system fails is equal to a wrong response [3]. The two characteristics which decide the reliability of cloud systems are timeliness and fault tolerance.

Although most of the current cloud platforms consider many of the failurechallenges , their implementation usually propose no fault tolerance solution ([4], [5]) or basic FT solutions ([6]). Most of the other solution provided in ([7], [8], [9], [10 entrust the responsibility of fault management either to the customer or to the provider rather than finding a reliable solution . The rest of the paper is organized as follows. Sections II discusses the related works and III covers fault tolerance in the cloud. Section IV discusses Fault Tolerance Techniques. Section VI concludes the work.

II. RELATED WORK

As we mentioned in Section I, few works addressed the issues of fault tolerance in cloud environments. Some platforms such as Eucalyptus [13] or CLEVER [14] provide no solution to take into account hardware, VM or customer application failures. CLEVER addresses FT management, but only for its own components.

Hadoop [22] was inspired by Google's MapReduce and Google File System (GFS) [23, 24] which provided access tothe file systems supported by Hadoop. Hadoop cluster will include a single master and multiple worker nodes. The master node consists of a JobTracker, task tracker, NameNode, and DataNode. The Hadoop Distributed File System (HDFS) usesit during replicating data and tries to keep different copies dataon different racks. The goal is to reduce the impact of a rackpower outage or switch off failure. Thus, even when these events occur, the data may still be readable. However, it takes along time to restart the system when failure occurs.

Chao Tung Yang et al propose a Distributed Replicated Block Device technology for reducing minimal down time and thereby avoiding failures [25]. A mechanism to reach Hadoop High Availability which called Virtualization Fault Tolerance (VFT) is proposed by the authors .

Author [26] has proposed the model which incorporates that system tolerates the faults and makes decision on the basis of reliability of the processing nodes i.e. VMs. According to his model, VM reliability is adaptive which changes after every computing cycle. If a VM produce a correct result within the time limit, it's reliability increases and if it fails to produce a result within time or correct result, it's reliability decreases. If the node continues to fail, it is removed and a new node is added.

III. BASIC CONCEPTS:

Fault Tolerance at Customer Level and the Cloud Provider Level

Application failures are identified at the customer level. The policy of fault detection depends on the application. The mechanism used to implement a detection policy is generally the same. The customer uses sensors which are deployed as software components in each application that is used to monitor the liveliness of the application .When there is a malfunctioning or repair the sensor trigger the fault.

At the cloud provider level, a VM FT technique can be implemented to deal with Fault Tolerance. As the provider can have direct access to the virtual machine hypervisor, he can implement it directly. Such an implementation decreases the number of VM sensors (and their associated communication) as they are integrated in hypervisors.

Replication types

a. Primary-backup replication:

In the primary-backup strategy [11], one of the replicas, called the primary receives the invocations from the client process, and sends the response back. Given an object x, its primary replica is noted prim(x). The other replicas are called the backups. The backups interact with the primary, and do not interact directly with the client process

b. Active replication

In the active replication technique, also called state-machine approach" [12], all replicas play the same role: there is here no centralized control, as in the primarybackuptechniqueas mentioned in Figure [1]

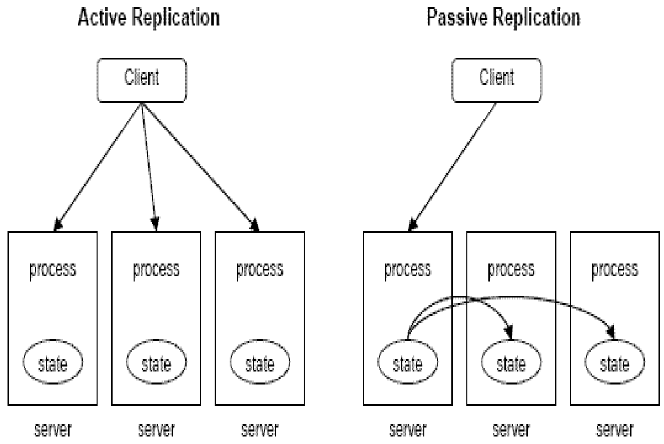

Figure 1.Active and Passive Replication

c. Passive Replication:

In **passive replication** there is only one server (called primary) that processes client requests. After processing a request, the primary server updates the state on the other (backup) servers and sends back the response to the client. If the primary server fails, one of the backup servers takes its place. Passive replication may be used even for non-deterministic processes. The disadvantage of passive replication compared to active is that in case of failure the response is delayed.

VM Fault Tolerance

Fault Tolerance provides continuous availability for virtual machines by creating and maintaining a Secondary VM that is identical to, and continuously available to replace, the Primary VM in the event of a failover situation.

Fault Tolerance can be enabled formost of the mission critical virtual machines. A duplicate virtual machine, called the Secondary VM, is created and runs in virtual lockstep with the Primary VM. The lock up set up captures inputs and events that occur on the Primary VM and sends them to the Secondary VM, which is running on another host. Using this information, the Secondary VM's execution is identical to that of the Primary VM. Because the Secondary VM is in virtual lockstep with the Primary VM, it can take over execution at any point without interruption, thereby providing fault tolerant protection.

IV. FAULT TOLERANCE TECHNIQUES

A large number of studies have conducted on the data fault-tolerant systems in the recent years that resulted in the invention of new strategies for finding the advantages and obstacles of fault-tolerant systems. In this section, we introduce the most recent fault-tolerance techniques in the Cloud Computation.

Fault masking: The fault masking is a structural redundancy technique to correct faults immediately for any kind of hardware redundancy. It completely masks the set of redundant modules. A set of similarmodules execute the same functions and the output is voted to remove errors created by a faulty module called as error voting.

Reconfiguration:The ability of a system to alter the active interconnection among modules, has a history of different purposes and strategies. Its purposes develop from the relatively simple desire to formalize procedures that all processes have in common to reconfiguration for the improvement of fault-tolerance, to reconfiguration for performance enhancement, either through the simple maximizing of system use or by sophisticated notions of wedding topology to the specific needs of a given process. The following are some of the applicable reconfiguration approaches Fault detection, Fault location, Fault containment, Fault recovery.

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 .

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.

V FAULT TOLERANCE MODELS IN CLOUD ENVIRONMENT

1. **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].
2. **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.
3. **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 [18].
4. **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 [17].

Table 1 gives a detail picture about the various fault tolerance techniques

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[18]	Dead line of work flow	Replication and resubmission of jobs
FTM[17]	Reliability, availability, on demand service	Replication users application and in the case of replica failure use algorithm like gossip based protocol.
CANDY[20]	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[21]	Usability, security, scaling 1	1. Two layer authentication and standard technical solution for the application.

5.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 [20].

6. 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 [21].

1. **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].

V. CONCLUSION

This study surveyed from fault tolerance focus on distinct functional areas to progress in making grid system more reliable. In order to provide reliable grid systems, it includes software resources, user applications, checkpointing, scheduling, Agents strategies, load balancing, and Workflows. Our survey discusses about all the above areas with different problems and some problems remain to be solved. Finally, Grid middleware's, self adaptive fault tolerance framework, workflow management based Serviceoriented Architecture (SOA), SLA are more dependable and trustworthy in cloud environments.

REFERENCES:

[1] J. Stankovic, "Misconceptions About Real-Time Computing," IEEE Computer, Vol. 21, No. 10, October 1988, pp. 10-19.
[2] O. S. Unsal, I. Koren, M. Krishna, "Towards Energy-Aware Software Based Fault Tolerance in Real Time Systems" ISLPED '02, Monterey, California, USA, August 12-14, 2002,
[3] K. H. Kim, "Towards Integration of Major Design Techniques for Real-Time Fault-Tolerant Computer System", Society for Design & Process Science, USA, 2002
[4] C. G. . Daniel Nurmi, Richard Wolski, Chris Grzegorczyk, GrazianoObertelli, Sunil Soman, Lamia Youseff, and DmitriiZagorodnov"The eucalyptus open-source cloud-computing system," in 9th International Symposium on Cluster Computing and the Grid (CCGRID). Washington, DC, USA, 2009.
[5] U. of Chicago, "Nimbus is cloud computing for science," http://www.nimbusproject.org/.
[6] OpenNebula, "Opennebula.org: The open source toolkit for cloud computing," http://opennebula.org.
[7] VishonikaKaushal and VishonikaKaushal, "Autonomic fault tolerance usinghaproxy in cloud environment," International Journal of Advanced Engeneering Sciences and Technologies, vol. 7, 2010.
[8] Microsoft, "Windows azure: Microsoft's cloud services platform," http://www.microsoft.com/windowsazure/.
[9] F. Tusa, M. Paone, M. Villari, and A. Puliafito, "Clever: A cloudenabled virtual environment," in IEEE Symposium on Computers and Communications (ISCC).Riccione, Italy, 2010.
[10] Amazon, Inc, Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: http://aws.amazon.com/ec2/#pricing
[11] N. Budhiraja, K. Marzullo, F.B. Schneider, and S. Toueg.The Primary-Backup Approach. In SapeMullender, editor, Distributed Systems, pages 199{216. ACM Press, 1993.
[12] F.B. Schneider. Replication Management using the State-Machine Approach. In SapeMullender, editor, Distributed Systems, pages 169{197. ACM Press, 1993.
[13] C. G. . Daniel Nurmi, Richard Wolski, Chris Grzegorczyk, GrazianoObertelli, Sunil Soman, Lamia Youseff, and DmitriiZagorodnov "The eucalyptus open-source cloud-computing system," in 9th International Symposium on Cluster Computing and the Grid (CCGRID). Washington, DC, USA, 2009.
[14] F. Tusa, M. Paone, M. Villari, and A. Puliafito, "Clever: A cloudenabled virtual environment," in IEEE Symposium on Computers and Communications (ISCC). Riccione, Italy, 2010.
[15] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper 1st Edition, **(2009)**.
[16] 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).
[17] R. Jhawar, V. Piuri and M. Santambrogio, "A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing", IEEE.
[18] W. B. Zhao, P. M. Melliar and L. E. Mose, "Fault Tolerance Middleware for Cloud Computing", IEEE 3rd International Conference on Cloud Computing, **(2010)**.
[19] G. Vallee, K. Charoenpornwattana, C. Engelmann, A. Tikotekar and S. L. Scott, "A Framework for Proactive Fault Tolerance".
[20] 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)**.
[21] 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)**.
[22] Hadoop. Available: http://hadoop.apache.org
[23] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M.Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A Distributed Storage System for Structured Data," ACM Trans. Comput.Syst., vol. 26, pp. 1-26, 2008.
[24] S. Ghemawat, H. Gobioff, and S.-T.Leung, "The Google file system," SIGOPS Oper. Syst. Rev., vol. 37, pp. 29-43, 2003.
[25] Chao Tung Yang, Dept. of Comput. Sci., Tunghai Univ., Taichung, Taiwan ; Wei-Li Chou ; Ching-Hsien Hsu ; Cuzzocrea, A.On Improvement of Cloud Virtual Machine Availability with Virtualization Fault Tolerance Mechanism, cloud Computing Technlogy and Science,IEEE Conference ,Nov 2011.
[26] Sheheryar Malik and FabriceHuet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Services.