

Proactive Fault Tolerance for Resilience Cloud Data Centers to Improve Performance Efficiency

Dr. Madhu B. K

Professor & Head of Department of Information Science Engineering,
R.R. Institute Of Technology, Bangalore

Ghamdan. M. Q

Research Scholar in Computer Science Department,
Jain University, Bangalore

Abstract:- Scaling failures are increasing in cloud Data Centers. The current state of practice of dealing with failure is not capable of dealing with failures in the same efficient way as before. A fault tolerance framework that performs environmental monitoring, event logging, parallel job monitoring and resource monitoring to analyze the virtual machine reliability and to perform FT through such preventative actions can be proposed to find a solution for fault tolerance. Proactive prediction of Virtual Machine (VM) faults needs to be improved to reduce the down time and cope up the growing clouds.

I .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. Cloud Computing has gained substantial popularity in modern computer science. It provides solution for business applications. 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 computers increase in scale and become more complex in the architecture, it is not surprising that failures occur more frequently. In this case a long running application on a cloud server system will experience frequent interruptions and will be rolled back to start the computation all over again or from the last checkpoint. Handling these kinds of failures is a great challenge. 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 [11]. So the reliability depends not only on the logical result, but also

the time of delivery [17]. Failure to respond when a system fails is equal to a wrong response [12]. The two characteristics which decide the reliability of cloud systems are timeliness and fault tolerance.

Fault tolerance is a challenging issue in cloud data centers. Fault Tolerance (FT) is a mechanism or approach to handling failures. Failures are increasing and it is important to improve the fault tolerance. Hosseini et al in [10] stated that the aim 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 it detects the suspicious items and replaces it with the correct data, which means discovering 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. A new approach may be proposed to deal with failures in a proactive way instead of waiting for failures to occur and react to these failures. The proactive approach requires a failure to be predictable. After a failure is predicted a decision is made to do a migration from a deteriorating node to a spare node. Therefore a mechanism for handling failures in cloud systems to improve reliability, availability and serviceability is needed to be framed.

Although most of the current cloud platforms consider many of the failure challenges, their implementation usually propose no fault tolerance solution [3], [20] or basic FT solutions [18].

Most of the other solutions provided in [21], [15], [7], [1] 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 discuss the related works and III covers the literature review on fault tolerance in the cloud. Section IV discusses basic concept related to Fault Tolerance Techniques. Section V 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 [3] or CLEVER [7] provide no solution to take into account hardware, VM or customer application failures. CLEVER addresses FT management, but only for its own components.

Hadoop [9] was inspired by Google's MapReduce and Google File System (GFS) [6], [19] which provided access to the 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) uses it during replicating data and tries to keep different copies data on different racks. The goal is to reduce the impact of a rack power outage or switch off failure. Thus, even when these events occur, the data may still be readable. However, it takes a long 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 [4]. A mechanism to reach Hadoop High Availability which called Virtualization Fault Tolerance (VFT) is proposed by the authors. Author [14] 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 produces a correct result within the time limit, its reliability increases and if it fails to produce a result within time or correct result, its reliability decreases. If the node continues to fail, it is removed and a new node is added.

Fault Tolerance according to Ganga, K. et al. [8] is a major concern to guarantee the 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 tolerant techniques are used to predict the failure in appropriate action. Fault tolerance is one of the important key issues in cloud. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics etc. When multiple instances of an application are running on several virtual machines and one of the servers goes down, there exists a fault and it is implemented by fault tolerance.

Proactive Fault Tolerance: It refers according to [2], [13] [8] to avoiding failures, errors and faults by predicting them in advance and proactively replaces the suspected components by other working components thus avoiding recovery from faults and errors. Some of the techniques which are based on these policies are preemptive migration, software rejuvenation, self healing etc as shown in figure (1).

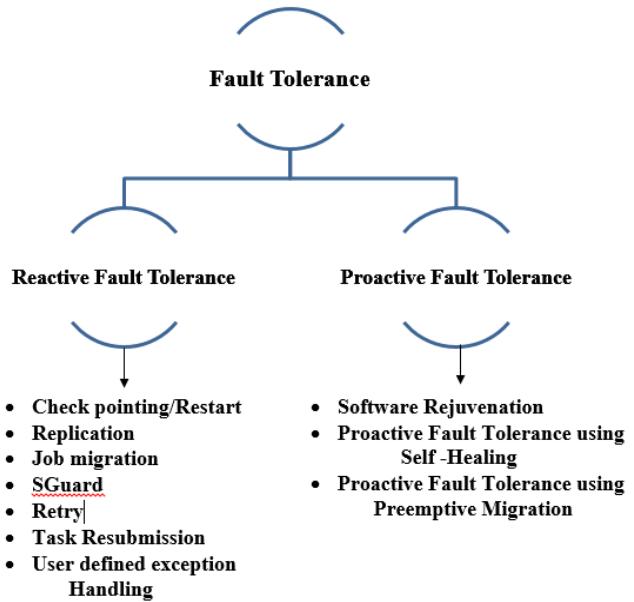


Figure 1

- Preemptive Migration: Proactive fault tolerance using preemptive migration relies on a feedback loop control mechanism where application is constantly monitored and analyzed as in figure (2).
- 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.
- Software Rejuvenation: It is a technique that designs the system for periodic reboots. It restarts the system with clean state.

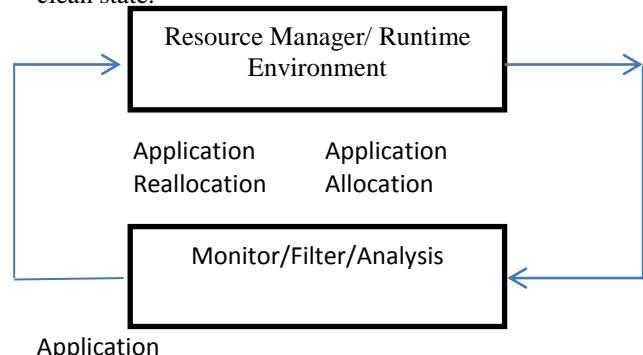


Fig (2) Feedback-loop control of proactive FT using preemptive migration

III. LITERATURE REVIEW

Hadoop [11] was inspired by Google's MapReduce and Google File System (GFS) which provided access to the 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) uses it during replicating data and tries to keep different copies data on different racks. The goal is to reduce the impact of a rack power outage or switch off failure. Thus, even when these events occur, the data may

still be readable. However, it takes a long 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 [3]. A mechanism to reach Hadoop High Availability which called Virtualization Fault Tolerance (VFT) is proposed by the authors.

OpenVZ [20], is a container-based virtualization for Linux to achieve fault tolerance .OpenVZ creates multiple secure and isolated containers on a single physical server enabling better server utilization and ensuring that applications do not conflict.

Author [21] 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.

A Hypervisor-based fault tolerance (HBFT), which synchronizes the state between the primary VM and the backup VM at high frequency of tens to hundreds of milliseconds is proposed by Jun Zhu in [15] . Based on the behavior of memory accesses among check pointing epochs, two optimizations called read-fault reduction and write-fault prediction, for the memory tracking mechanism is introduced . These two optimizations improve the performance of application by 31 percent and 21 percent. Then, a software super page which efficiently maps large memory regions between virtual machines (VM) is proposed for proactive fault tolerance.

Alain Tchana et al [7] propose a fault tolerance method in which both the Cloud customers and providers will collaboratively share their responsibilities in order to provide required fault tolerance. According to Tchana, application faults can be detected and repaired at the customer level. But the Virtual Machine and Hardware faults can be detected and repaired at the Cloud provider level. The recovery/restoration of the applications running on the refurbished VMs can be requested and performed at the customer level. Checkpointing technique is used to create restore points for the recovered VMs.

Wenbing Zhao et al [1] put forward a Low Latency Fault Tolerance (LLFT) middleware framework for fault tolerance which replicates the processes of applications, using the leader/follower replication approach. This framework is equipped with a LLFT Messaging Protocol which ensures reliable communication between the replicated processes. LLFT Membership Protocol ensures that the entire replicated process group has a consistent view of their membership.

Sheheryar Malik et al [16] propose a fault tolerance model for real time Cloud Computing. In this model, the faults are managed based on the reliability of processing nodes or virtual machine. According to authors, the reliability of nodes changes in every computational cycle. The proposed fault tolerance model collects and analyses

the performance or reliability metrics of a particular virtual machine. If a particular VM can produce the correct results within the speculated time, that node or VM is considered to be worthy node and its reliability increases. There is a minimum value for reliability for which a particular node is to be considered worthy or fault tolerable VM. And if a node fails to produce the minimum result within the specified time, its reliability decreases and the system undergoes backward recovery or safety measure.

Pranesh Das et al [5] propose a Virtualization and Fault Tolerance (VFT) technique by increasing the system availability and reducing the service time. This reactive fault tolerant technique consists of a Cloud Manager (CM) module and a Decision Maker (DM) which are used to manage the virtualization, load balancing and to handle the faults. The first step involves virtualization & load balancing and in the second step fault tolerance is achieved by redundancy, checkpointing and fault handler. The virtualization includes a fault handler. Not all the faults are recoverable. Fault handler finds these unrecoverable faulty nodes and restricts these virtual nodes from future requests or usage. It also helps to remove the temporary software faults from recoverable nodes making them available for future requests.

The fault tolerance algorithm proposed in [3] supports nested object invocations. The chief advantages of the scheme are: a) No action is needed in the case of failure of a secondary replica; b) The time to recover from a primary failure is minimal, c) Separation of replication protocol and reliable communication protocol. To recover from a primary failure the system need to (detect the failure and) select one of the secondaries to become the primary. The designated secondary can become primary once it has made sure that its current state is equivalent to the state of the failed primary (it can do so by processing outstanding requests, if any). This is in contrast with the checkpointing and rollback recovery scheme, where the recovery time can be substantial.

Alexandru Costan, CiprianDobre [7] propose an approach relying on replication techniques and based on monitoring information to be applied in distributed systems for fault tolerance. The proposed approach uses both active and passive strategies to implement an optimistic replication protocol. Using a proxy to handle service calls and relying on service replication strategies. The complexity and overhead issues are dealt with. The authors present architecture for implementing the proxy based on monitoring data and the replication management. Experimentation and application testing using an implementation of the architecture is presented. The proposed architecture is demonstrated to be a viable technique for increasing dependability in distributed systems.

ANJU BALA et al. [2] put forward an idea of designing an intelligent task failure detection models for facilitating proactive fault tolerance by predicting task failures for scientific workflow applications. The working of model is distributed in two modules. In first module task failures are predicted with machine learning approaches and in second module the actual failures are located after

executing workflow execution in cloud test-bed. Machine learning approaches such as naïve Bayes, ANN, logistic regression and random forest are implemented to predict the task failures intelligently from the dataset of scientific workflows. The thesis proposes the use of artificial Neural Networks for detecting the faults in cloud environment. The faults are first detected and then suitable fault tolerance technique (pre-emptive migration/ checkpointing) is applied to make the system fault tolerant. The faults will be handled proactively and this will help to resolve the problems associated with fault tolerance techniques.

IV. 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 liveness 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.

Advantage of Proactive Fault Tolerance

Different technologies from competing vendors of cloud infrastructure need to be integrated for establishing a reliable system. Therefore autonomic fault tolerance must react to synchronization among various clouds is required. If a virtual machine fails due to hardware problems a mechanism to rapidly detect and respond to hardware failure is required so that virtual servers can instantly be moved to an alternate host. This is made possible by proactive fault tolerance. The basic premise of Proactive fault tolerance in cloud data centers is that a primary VM and a secondary VM are kept in perfect sync. That way, if the primary VM fails, the secondary VM is ready to take over in an instant.

Technique used for proactive fault tolerance is:

MapReduce Fault Tolerance : MapReduce has been gaining popularity and it has been used at Google extensively to process 20 peta bytes of data per day. Yahoo developed its open-source implementation, Hadoop, which is also used in Facebook for production jobs including data import, hourly reports, etc. As shown in fig.4, in MapReduce, input data is split into a number of blocks, each of which is processed by a map task. The intermediate data files produced by map tasks are shuffled to reduce tasks, which generate final data output [13].

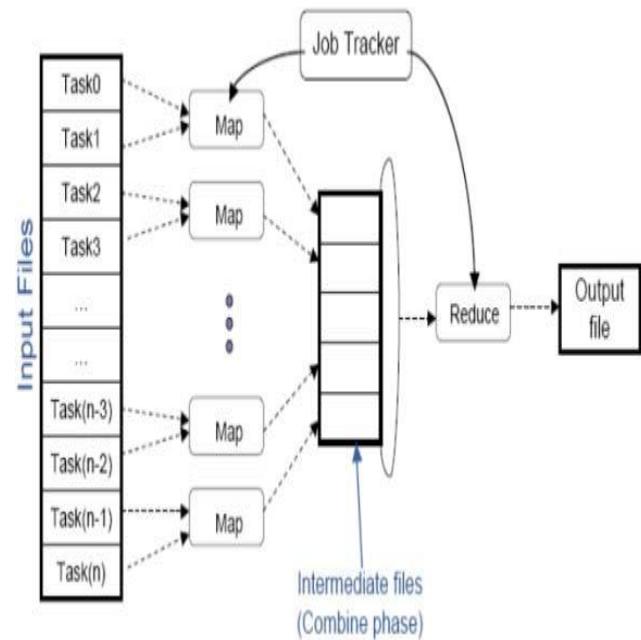


Fig.4 MapReduce Architecture

MapReduce handles failures through re-execution. If a machine fails, MapReduce reruns the failed tasks on other machines. In the effect of a single machine failure on the runtime of a Hadoop job (i.e., a two-stage job) is studied and it is found out to cause a 50% increase in completion time.

V. CONCLUSION:

There are number of fault tolerance models which provide different fault tolerance mechanism to enhance the system performance and reliability. There are some drawbacks with the existing fault tolerance models. A possibility to overcome the drawbacks of all previous models and try to make a reliable model which will reduce down time and able to cover maximum fault tolerance aspects in cloud data centers is the crux of this proposed research work.

Finally we can conclude that yet there is no standard metric and interface for Fault Tolerance. Each solution uses its own metric for evaluating health and performance of virtual machines which makes comparison and integration very difficult. So a common standard metric for predicting fault tolerance is required as there are multiple services provided involved in a single service.

REFERENCE:

- [1] Amazon, Inc, Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: <http://aws.amazon.com/ec2/#pricing>.
- [2] Amin, Zeeshan, Harshpreet Singh, and Nisha Sethi. "Review on fault tolerance techniques in cloud computing." International Journal of Computer Applications 116.18 (2015).
- [3] 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.

[4] 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 Technology and Science ,IEEE Conference ,Nov 2011.

[5] F.B. Schneider. Replication Management using the State-Machine Approach. In SapeMullender, editor, Distributed Systems, pages 169{197. ACM Press, 1993.

[6] 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.

[7] F. Tusa, M. Paone, M. Villari, and A. Puliafito, "Clever: A cloud enabled virtual environment," in IEEE Symposium on Computers and Communications (ISCC).Riccione, Italy, 2010.

[8] Ganga, K., S. Karthik, and A. Christopher Paul. "A Survey on Fault Tolerance in Work flow Management and Scheduling." International Journal of Advanced Research in Computer Engineering & Technology 1.8 (2012): 176-179.

[9] Hadoop. Available: <http://hadoop.apache.org>.

[10] Hosseini, Seyyed Mansur, and Mostafa Ghobaei Arani. "Fault-Tolerance Techniques in Cloud Storage: A Survey." International Journal of Database Theory and Application 8.4 (2015): 183-190.

[11] J. Stankovic, "Misconceptions About Real-Time Computing," IEEE Computer, Vol. 21, No.10, October 1988, pp. 10-19.

[12] K. H. Kim, "Towards Integration of Major Design Techniques for Real-Time Fault-Tolerant Computer System", Society for Design & Process Science, USA, 2002.

[13] Kaur, Jasbir, and Supriya Kinger. "Analysis of Different Techniques Used For Fault Tolerance." IJCSIT: Int J Comput Technol 4.2 (2013): 737-741.

[14] Malik and FabriceHuet, Sheheryar. "Adaptive Fault Tolerance in Real Time Cloud Computing." 2011 IEEE World Congress on Service.

[15] Microsoft, "Windows azure: Microsoft's cloud services platform," <http://www.microsoft.com/windowsazure/>.

[16] 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.

[17] 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.

[18] OpenNebula, "Opennebula.org: The open source toolkit for cloud computing," <http://opennebula.org>.

[19] S. Ghemawat, H. Gobioff, and S.-T.Leung, "The Google file system," SIGOPS Oper. Syst. Rev., vol. 37, pp. 29-43, 2003.

[20] U. of Chicago, "Nimbus is cloud computing for science," <http://www.nimbusproject.org/>.

[21] VishonikaKaushal and VishonikaKaushal, "Autonomic fault tolerance using haproxy in cloud environment," International Journal of Advanced Engeneering Sciences and Technologies, vol. 7, 2010.