International Journal of Engineering & Technology, 7 (2.33) (2018) 67-70

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Review on failure forecast in cloud for a fault tolerant system

J. M. Nandhini [1] *, T. Gnanasekaran [2]

[1] *Research Scholar, Anna University, Asst Prof, Sri Sai Ram Institute of Technology, Chennai*
[2] *Professor , R.M.K Engineering College, Chennai*
**Corresponding author E-mail: jmnandhinisaravanan@gmail.com*

Abstract

Cloud Computing is an increasingly popular computer paradigm constituting a large infrastructure involving storage, memory, servers and applications accessible via computer network. The cloud system design aims to provide on-demand services with scalability on diverse resources to ensure efficient resource utilization in addition to effectiveness. As cloud is a service-oriented infrastructure, it is critically imperative that the system is highly reliable to meet the Service Level Agreement (SLA). To achieve reliability, cloud requires a very efficient fault tolerance mechanism. Serviceability and reliability is impacted by any failure in the system. Prior prediction of faults in the system helps in overcoming failures. The Fault Tolerance in cloud involves ascertaining the resource fitness to execute scheduled task. The process involves prior screening of resources against various tasks as part of scheduling process. The scheduling process relies significantly on the virtualization of resources to maintain high efficiency.

Keywords: *Fault Tolerance; Virtualization; Checkpoint; Fault Prediction*

1. Introduction

Cloud Computing is a highly developed computing environment emerged from utility computing and grid computing. It can be defined as service delivered over a network rather than physically having the computing resources at the location. A scheduler acts an interface between the users and the resources. The possibilities of a resource failure are higher than in parallel computing which affects the job execution. Fault Tolerant systems reveal the capability of the system to abide to the faults which results in failure of the system. Fault tolerance can be measured through reliability and availability. Availability is directly proportional to the cloud infrastructure, which can detect the faults and eradicate it. The duration for fault identification and its elimination should be minimal as it affects the Qos. The process becomes complex for a non virtualized data centre. To avoid the complexity, the data centre can be virtualized; identification of faults can be predicted at the earliest.
In this paper, a prediction technique for fault identification and task prediction in cloud system is proposed during task execution with several resources.

2. Implication of fault tolerance

Fault Tolerance facilitates to maintain and provide the services as per requirement regardless of the faults in the system. It intends to avert the failures despite the existence of faults. Since cloud uses a shared pool of configurable resources, fault tolerance is important to ensure its correct behavior. The fault tolerance service is necessary to satisfy the QOS requirements. Various types of resource failure including network failure, process failure are dealt .These failures in the system will affect the timing deadline and the Qos. [3]. Fault tolerance is an approach where a system continues to succeed even if there is a fault. It also assures the reliability and availability of the application. Failures should be managed to reduce its impact over the system.

2.1. Faults and failures

A system is said to fail when it fails to complete its task within its expected deadline. An error in the system may lead the failure. The source of an error is termed as fault. It the system deviates from its specifications or it is unbale to deliver the output within the estimated time,it may said to be a failure. Neither a fault may not be the cause of an error nor an error to a failure. [1]

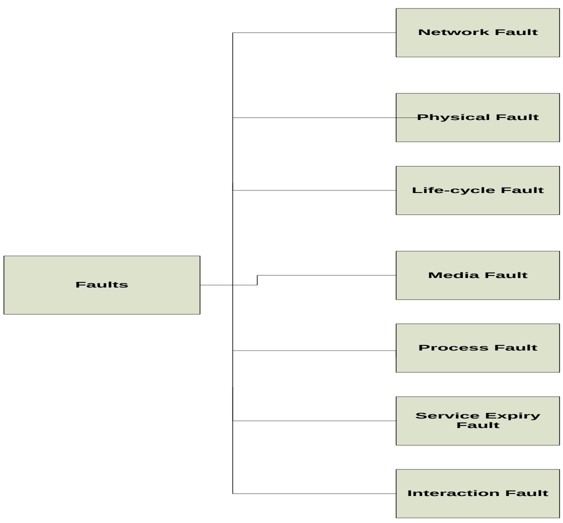

Fig. 1: Types of Faults.

The above figure shows the various types of faults caused based on several factors in a distributed system

The types of faults can be grouped under the following classification

1) Transient Fault: Transient Faults occurs once and disappears on repeated operation. An example is a Transmission of a network message to a specified destination may fail initially but succeeds on retransmission
2) Intermittent Fault: Intermittent Faults occurs disappears, and then reappears and so on. This fault is very difficult to diagnose. An example is a loose connection n a network
3) Permanent Fault: Permanent faults continuous to exist until the faulty components are fully replaced. Software bug, disk crash are some examples

   According to cloud architecture there are three types of failures can occur in a cloud environment. [6]
4) Hardware failure : Failure in hardware lead to the physical machine failure
5) VM failure : Failure in virtual machine in the physical host, produces in correct results
6) Application Failure: The executing of an application fails

3. Dependability

The fault tolerance makes the system more dependable. Dependability of the system can be relied to offer the services to which it has been bound to. The tasks to accomplish Dependability are

1) Fault prevention: The introduction of faults and their occurrences are prevented.
2) Fault Tolerance: It makes the system to function in an expected way despite of faults.
3) Fault Removal: It reduces the quantity and severity of faults.
4) Fault forecasting: It estimates the current number and the forthcoming occurrence of faults that are likely to occur [4]

Dependability can be calculated based on reliability and availability. Reliability is the capability of the system to execute its services properly on demand. The term availability implies the performance of the system for the service on request.

3.1. Reliability

Reliability specifies that the system can deliver its services constantly without any failure. Such systems can be treated as highly reliable which offers services with no disruption over a long period of time. It can be defined as the probability that the system would execute the tasks on time. It is closely associated with the terms mean time to failure (MTTF) and mean time between failures (MTBF). Taking into consideration the mean time to repair (MTTR), MTTF can be calculated as

$$MTBF = MTTF + MTTR$$

3.2 Availability

Availability implies that a system is immediately ready for use. It is defined as the likelihood of the system to operate at desired time. This attribute is reliant on the time period that it consumes to come back to its service on failure.

$$Availability = (MTTF) / (MTTR + MTTF)$$

4. Fault tolerance techniques

In the view of task flow and work flow fault tolerance in cloud computing can be classified into two categories. The fault tolerant techniques is shown in Figure 1

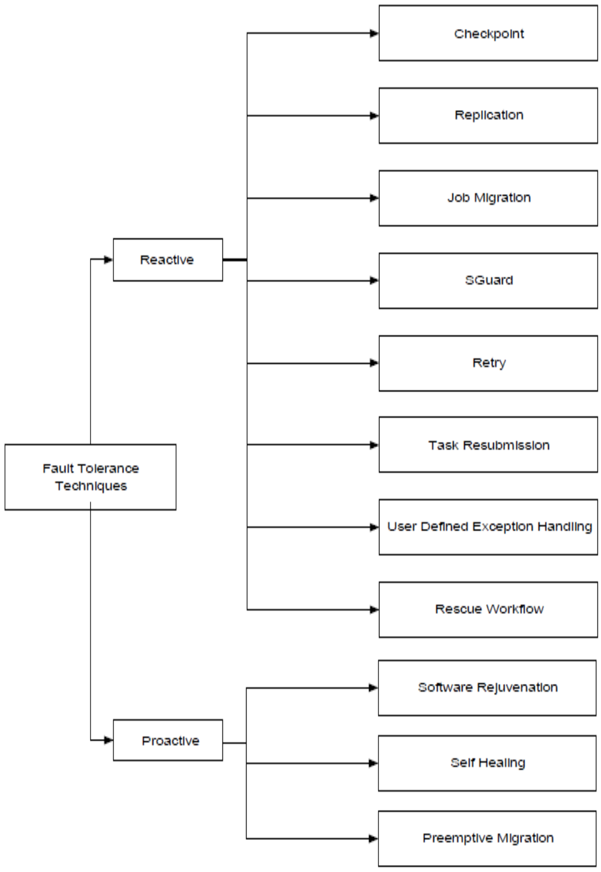

Fig. 2: Fault Tolerance Techniques.

4.1. Reactive fault tolerance

This technique eliminates the faults after its occurrence. In addition, reduces the impact of the failures on the system. This can be executed by the following schemes. [4].

- Checkpoint: Which the system can be restored on failure thereby reducing the amount of re-computations concerned from the previous failure from the saved state.
- Replication: User access is eased by replicating prioritized data at several suitable locations.
- Job Migration: The process of migrating a failed job in a virtual machine to another virtual machine where it resumes its execution. HAProxy uses this approach.
- SGuard: This method uses rollback recovery. Examples of this method are HADOOP and Amazon Ec2.
- Retry: The failed task is again resubmitted on the same cloud resource.
- Task Resubmission: In this approach, the failed task is again submitted to the same VM or to another VM during its execution
- User defined exception handling: It has a predefined workflow for execution when the tasks fail.
- Rescue workflow: The system continues to proceed in its execution, even in account of failure until it cannot progress more without fixing the fault.

4.2. Proactive fault tolerance

Proactive fault tolerance avoids recovery by calculating the probable events of faults by forecasting the failures. This can be implemented by the following schemes:

- Software Rejuvenation: In this scheme, the application is started as a clean state. The rejuvenation intervals can be calculated periodically at intervals for the application to have a clean start.
- Self-Healing: This scheme suits well the situation, where a single application having more than one instance running at

different virtual machines. The failure is automatically handled, if case of any fault in the VMs.

- Preemptive Migration: It migrates the task to another VM, if the current node on which the application runs is about to fail.

5. Forecasting fault in task scheduling

The resource requirements in terms of CPU, RAM, and Disk Space, the scheduling priority and constraints are associated with every task. The job of a good scheduler is to improve the maximum utilization of allocated resources thereby minimizing the execution time. A task runs for a finite time on behalf of an application. They are designed to run on their own containers and uses minimal resources. The state of a task can be obtained. Many task-scheduling algorithms are proposed, but still cloud schedulers meet many failures owing to unanticipated events and unexpected demands of service. Hence, an efficient task scheduling in need to build a proactive response to the changes incurred in cloud environment. Prediction of changes may lead to the modification of scheduling decision so as to minimize the task scheduling failures. There is a close correlation between job failure and task failure. Tasks failure directs the job failure. Forecast the task failure and killing those tasks would preserve the resource wastage but fails to offer a good QoS.

A good scheduler can minimize the execution time and improve the utilization of the allocated resources [5]. The core objective of the scheduler is to trace a optimal solution for the submitted jobs to the appropriate virtual machines in concurrence with the best possible execution time and its resource availability. The scheduler thereby minimizes the resource utilization cost by choosing the virtual machine to process the tasks to meet the requirements.

Unlike other algorithms, cloud computing contains the virtualization layer, which comes up with an extra setup in scheduling. For scheduling the resources in cloud, the scheduler has to decide upon two mapping issues. The first issue is the mapping between the VM and the host and the second issue is the mapping between the task and the VM.

6. Related work

Many algorithms have been proposed for handling failures in cloud and making the system as fault tolerant. Most of these algorithms work uses the reactive techniques. The most commonly used reactive technique is checkpoint and replication. In checkpoint, the current state of the system is saved at the checkpoint interval. In case of any fault, the system is restored from the state that has been previously saved and resumes as normal execution. A dynamic adaptive checkpoint is used for fixing the checkpoint dynamically based on failure rate of VM. The replication strategy is used where the application runs on more on one VM's. Even if any one of the VM's fails, the system will be able to meet the deadline with other VM's.

In [7], the coordination between the VM's has been taken as a major issue. The authors have implemented both the techniques at two levels. As a first level, the proactive technique is used to exploit the VM status. Based on the failing status of the VM, a coordinated checkpoint is used to save the current status of the task. To predict the fading VM Statistical metric model is used. An efficient heuristic algorithm is used to choose the optimal target host.

In [8] a method has been employed using the proactive technique, software rejuvenation technique of fault tolerance. This paper implements the technique at two stages. The first stage involves the failure detection and aging degree is evaluated which predicts the rejuvenated cloud services. In the second stage, architecture is modeled that consists of different cloud services and loosely and tightly coupled components running on virtual machines.

Message log is a vital tool for fault prediction and detection. In [9], two modules namely fault prediction and fault detection is involved in determining the faults in the datacenter. Pattern dictionary provides message pattern and probability to predict faults and a probabilistic model is proposed based on the parameters CPU idle time, memory usage for fault detection.

A comparative study of the architecture model down with its methods is shown below.

Table 1: Comparative Study of Architecture Model

Architecture Model	Proposed by	Fault Tolerant Technique	Factors considered
Adaptive Fault Tolerance in Real time Cloud Computing (AFTRC)	Sheheryar Malik et al	Adaptive	Computing Node Reliability
Dynamic Adaptive Fault Tolerance (DAFT)	Sun et al	Adaptive	Checkpoint overhead, fault overhead
Fault Tolerance Manager (FTM)	Ravijawhar et al	Reactive	Introduction of dedicated Layer
Dynamic Clustering League championship algorithm (DCLCA)	ShafiMuhammad Abdulhamid et al	Reactive	Makespan time, Failure ratio, Failure slowdown
Fault Tolerance for HPC systems	Ifeanyi et al	Proactive	Message passing interface
Framework for Fault Tolerance	G.Vallee et al	Proactive	Fault Tolerance Daemon implements proactive mechanism

7. Conclusion

A robust fault tolerance has become an indispensable part in cloud to provide high reliability and availability. The fault tolerance starts with task scheduling as an initial stage. Anticipation of failures at this initial stage can make the system more reliable and robust. This paper discusses the significance of fault tolerance including types of faults, techniques of fault tolerance with an emphasis on fault prediction during task scheduling and a comparative study on the diverse architectural models.

References

[1] https://arxiv.org/pdf/1507.03562.pdf.

[2] Paul Townend, Jie Xu, Fault tolerance within a grid environment, as part of the e-Demand project at the University Of Durham, DH1, United Kingdom, 2003.

[3] P. Latchoumy and P. Sheik Abdul Khader , "Survey On Fault Tolerance In Grid Computing", in International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011.

[4] Bala, A., & Chana, I. (2012)." Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing", International Journal of Computer Science Issues (IJCSI), 9(1).

[5] N. Shahapure and P. Jayarekha, \Load balancing with optimal cost scheduling algorithm," in International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), pages 24-31, (2014).

[6] Alain Tchana, Laurent Broto, Daniel Hagimon. (2012). Approaches to Cloud Computing Fault Tolerance, 978-1-4673-1550-0/ 12, IEEE.

[7] PFT-CCKP: A Proactive Cloud Services Fault Tolerance Mechanism Jialei Liu Shangguang Wang, Ao Zhou, Fangchun Yang State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications Beijing, China.

[8] B Jing Liu, Jiantao Zhou and Rajkumar Buyya, Software Rejuvenation based Fault Tolerance Scheme forcloud.

[9] Purvil Bambharolia, Prajeet Bhavsar, Vivek Prasad, Failure Prediction and Detection in Cloud Datacenters, International Journal of Scientific & Technology Research Volume 6, Issue 09, September 2017.

[10] Shafi'i Muhammad Abdulhami, Muhammad Shafie Abd Latiff, Syed Hamid Hussain Madni, Mohammed Abdullahi, Fault tolerance aware scheduling technique for Cloud Applications .2015 IEEE 8th International Conference on Cloud Computing.

[11] Ifeanyi P. Egwutuoha, Shiping Chen, David Levy,Bran Selic, Rafael Calvo, A Proactive Fault Tolerance Approach to High Performance Computing (HPC) in the Cloud, 2012 Second International Conference on Cloud and Green ComputingIEEE 2012.
[12] Ravi Jhawar,, Vincenzo Piuri, Marco Santambrogio , A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing.
[13] Zhao, W., Melliar,Smith, P. M., & Moser, L. E. (2010, July). Fault tolerance middleware for cloud computing. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on (pp. 67-74). IEEE.
[14] Sheheryar Malik, Fabrice Huet IEEE World Congress on Services, Jul 2011, Adaptive Fault Tolerance in Real Time Cloud Computing.
[15] Geoffroy Vallee, Kulathep Charoenporn wattana, Christian Engelmann, AnandTikotekar, Stephen L. Scott," A Framework for Proactive Fault Tolerance".
[16] Dawei Sun,Guiran Chang,Changsheng Miao,Xingwei Wang, Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments, The Journal of Supercomputing.