International Journal of Latest Trends in Engineering and Technology
Special Issue NCRTCC 2017, pp.032-036
e-ISSN:2278-621X

ADAPTIVE FAULT TOLERANCE-BYZANTINE TECHNIQUE

P. Renukadevi.[1], J.M.Nandhini[2]

ABSTRACT: Cloud computing is become popular and important solution for building highly reliable applications on distributed resources. As internet is growing in an immense including its users, cloud computing with its unbelievable possibilities are in ease, Quality of service. The administration has turned into a guaranteeing figuring stage for both business and non-business computation customers. It is an adoptable technology as it offers integration of software and resources which are dynamically scalable. The cloud results in various unexpected faults and failures in dynamic environment.The ability of a system to react gracefully to an unexpected equipment or programming malfunction is known as fault tolerance.Failure should be assessed and handled effectivelyin order to achieve robustness and dependability in cloud computing.This paper deals with various fault tolerance measures and metrics have been proposed to increase fault tolerance ability of cloud.

Keywords: Fault tolerance, reliable, Quality of service.

I. INTRODUCTION

Cloud computing is an emerging platform which provides computing and storage to the end-users as a service. The cloud platform is divided into 3 domains classified into three domains: Application (software), platform and infrastructure. The cloud plays an important role in the field of business for the end-users. Cloud computing has established a distribution over a considerable extent acceptance in various domains such as research, business, health, e-commerce, agriculture and social life. As the cloud computing systems continue to grow in scale and complexity it is important to ensure the availability, stability and reliability in those systems. The primary reasons that can accelerate failures and faults in the large-scaled complex and dynamic environments are execution environments, frequent updates and upgrades, online repairs and intensive workload. The reliability of systemscan be easily compromised if the proactive measures are not taken to tackle against the possible failures emerging in cloud subsystems. To achieve reliability the cloud service providers adopt various mechanisms to implement fault tolerance at the system level.

Fault tolerance is a vital issue in cloud computing platforms and applications. The ability of a system to react gracefully to an unexpected equipment or programming malfunction is known as fault tolerance. Various fault detection methods and architectural models have been proposed to increase fault tolerance ability of cloud.

1.1 TYPES OF FAULTS

Cloud platform has 3 types of layers: hardware, VM's and applications. There will be a failure in any layer throughout the execution of application. According to the nature of the fault needed actions are taken. Some of the faults are explained below

1.1.1 TRANSIENT

Transient fault occurs for certain duration of time and fault appears only once. Then disappears after taking necessary actions. For example: sometimes on first attempt the network message can't reach its specified destination but after retransmission the message is reached to the correct destination successfully.

1.1.2 INTERMITTENT

Intermittent faults are the faults that are reoccurring again and again. This type of fault is very irritating type and mainly occur due to failure of any component or improper operation among the components.

1.1.3 PERMANENT

The permanent faults remains within the systems unless and until the faulty components are been fully replaced or repaired. Example: disk crashes.

[1] *Sri Sai Ram Engineering College*
[2] *Sri Sai Ram Engineering College*

The fault appear as a failure of resources in an typical cloud environment. Some of the resources that are used by the end-users are applications/ hardware storage. The most common faults in the cloud environment are

A) Byzantine failures

B) Crash failures

A) Byzantine failures

In this type of failure the system components fail in an absolute way causing the system to behave incorrectly in an unpredictable manner. The system processes an incorrect input request and produces inconsistent outputs.

B) Crash failures

The system components completely stop functioning or remain inactive during crash failures.

1.2 REDUNDANCY

The implementation of fault tolerance can be achieved by redundancy that involves in duplication of hardware and software components. If a component or a process fails the backup process or a component is available to take place of primary.

For example:

Stratus computers- If one machine fails then the duplication allows another machine of the pair to continue the computation process without any delay

Tandem computers- Uses many independent identical processors and redundant storage devices. The controllers provide automatic recovery in any software and hardware failures.

1.3 FAULT TOLERANCE VALIDATION

The fault tolerance /availability analysis is provided by the cloud service provider to the end-users. The main importance of the fault –tolerance validation is to ensure the SLAs are properly bound. The random probability factor is involved. So it is difficult to verify the fault-tolerant machine for meeting the reliability requirements. Fault Injection methodology is used for assessment of system reliability.

Fault injection is an important method to copy or imitate the occurrence of errors in acontrolled environment to make the necessary measurements.

The tools used for fault tolerance validation are:

- Hybrid Automated Reliability Predictor (HARP)
- System Availability Estimator (SAVE)
- Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE)
- UltraSAN– (University of Illinois, UIUC)
- DEPEND – (UIUC)

A) HARP

HARP is a Hybrid Automated Reliability Predictor. It provides a toolbox of the integrated programs. This is helpful for the users to customize their application in workstation or non-workstation environment

B) **SAVE**

SAVE stands for System Availability Estimator. SAVE is a package that is used for constructing and solving probabilistic models. Mainly it is used as an art tool during the design and configuration of the system.

C) SHARPE

SHARPE is a Symbolic Hierarchical Automated Reliability and Performance Evaluator .It is a tool used for specifying and analyzing performance, reliability and performs ability models

D) ULTRASAN

The software package for model-based evaluation of systems is the UltraSAN and represented as stochastic activity networks (SANs).

E) DEPEND

DEPEND is a functional simulation tool. It provides an integrated design and fault injection environment for system level dependability analysis.

1.4 FAULT TOLERANCE MEASURES

Fault tolerance can be used to measure the quality of being dependable or reliable of cloud system. Two major computing measures for fault tolerance systems are:

- **Availability**
- **Reliability**

1.4.1 AVAILABILITY

Availability is the ratio between the uptime and sum of the uptime and downtime of the system.
Availability can be quantified as:

$$\textbf{Availability} = \frac{\textbf{Uptime}}{\textbf{Uptime} + \textbf{Downtime}}$$

Uptime is the measure of the time a machine, typically a computer has been working and is available. **Downtime** is the opposite of the uptime.
Availability can also be measured as percentage of agreed service time and downtime of the system.

$$\textbf{Availability}(\%) = \frac{\textbf{AgreedServiceTime} - \textbf{Downtime}}{\textbf{AgreedServiceTime}} \times 100$$

AgreedServiceTime -The expected operational time of the system per month

1.4.1.1 METRICS OF AVAILABILITY

There are 3 types of metrics for availability they are as follows:

A) **Mean Time To Failure(MTTF)**
B) **Mean Time Between Failure(MTBF)**
C) **Mean Time To Repair(MTTR)**

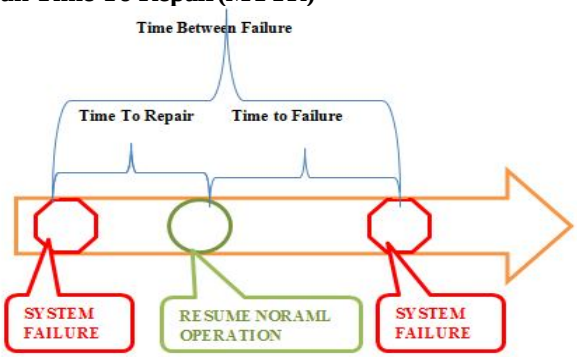

Figure 1:Metrics of availability

The MTTF is the average time of the system operating accurately until a failure occurs.
The MTBF is the average time between two consecutive failures of the system.
The MTTR measure predicts the average time required to replace the faulty component of the system to bring the system back to operational mode.

$$\textbf{Availability} = \frac{\textbf{MTBF}}{\textbf{MTBF} + \textbf{MTTR}}$$

1.4.2 RELIABILITY

Reliability is the probability of the system to work accurately.Reliability is also defined as "the ability of an item to perform a required function under stated conditions for a stated time interval". The Reliability of a service can be calculated as :

$$\textbf{Service Reliability} = \frac{\textbf{Successful Responses}}{\textbf{Total Requests}} \times 100$$

II. FAULT TOLERANT STRATEGIES IN CLOUD

Crash and failure are very common in the cloud.The effect of failure on a medium or long running jobs can lead to threatening the fullfillment of SLA contract and wastage of computation time.This problem can be handled by executing the backup process on different system for process. The fault-tolerant systems characterize the recovery from errors as either roll-forward or roll-back.

Roll forward mechanism takes the system state at the time when the error was detected in order to correct the error, and then the system moves forward. The roll-back mechanism utilize checkpointing to revert the system state to some earlier correct state, then the system moves forward from that particular state.

- **Checkpoint based Fault Tolerance**
- **Adaptive Fault Tolerance**

2.1 CHECKPOINT BASED FAULT TOLERANCE

The checkpoint mechanism records the state of the system periodically after certain time limit .If the failure occurs the last checkpoint state of the system is restored and the task execution is resumed from that point.

The checkpoint policy becomes more challenging in the virtualized environment where huge virtual machine(VM) images needs to be saved and restored.

2.2 ADAPTIVE FAULT TOLERANCE

Adaptive fault tolerance techniques help the system to maintain and improve the system's fault tolerance by adapting to the environmental changes.

The techniques for the cloud computing in adaptive fault tolerance, monitors the state of the system and reconfigure. The cloud computing system configurations for the stability of the system in case of error detections. One of the adaptive fault tolerance techniques is explained below:

2.2.1 BYZANTINE FAULT TOLERANCE CLOUD (BFTCLOUD)

The BFT cloud is an architecture that is designed for voluntary resource cloud computing. The voluntary resource cloud combines public resource computing and cloud compute infrastructure for distribute cloud that takes place of centralized system in the data center. The infrastructure consists of numerous user contributed resources. The computing resources are denoted as a node in the cloud environment. The improvement on reliability of cloud module under voluntary resource cloud infrastructure has been limited its traditional testing due to the following:

- The nodes that are contributed by the users are highly dynamic, much cheaper, less powerful and less reliable.
- There is not a reliable communication link between the modules.
- In voluntary resource cloud infrastructure nodes are connected by unpredictable communication links.

It is extremely critical to design a fault tolerance mechanism for handling different faults in order to build reliable cloud application.

Some Faults are:

- **Node Faults: Like Crashing**
- **Network Faults: Like Disconnection**
- **Byzantine Faults: Malicious Behavior**

Malicious Behavior is sending inconsistent response to the request. So traditional fault tolerance cannot tolerate malicious behavior of nodes .Therefore for this reason BFT Cloud is used. The BFT Cloud is based on Byzantine fault tolerance approach.

2.2.1.1 WORK PROCEDURE OF BFTCLOUD

The input of BFT Cloud is a sequence of request with specified QoS requirement sent by cloud module. Where the output of the BFT Cloud is a sequence of committed responses corresponding to request.

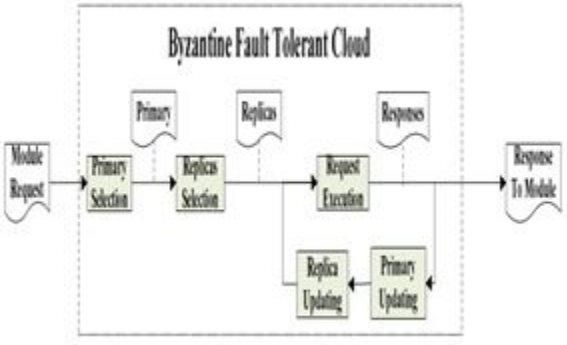

Figure 2: BFT Cloud working procedure

There are 5 phase's present they are as follows:

- **Primary Selection**
- **Replica Selection**
- **Request Execution**
- **Primary Updating**
- **Replica Updating**

(A) PRIMARY SELECTION:

Request for the service is handled in this primary selection. Selection of appropriate replicas for the group is done. Forwarding request to all replicas and replacing faulty replicas with the newly selected nodes. If the failure occurs in primary it decreases the overall performance.

(B) REPLICA SELECTION:

The set of nodes are selected as replicas. The Primary Selection forwards the request to all replicas for the execution.

(C) REQUEST EXECUTION:

The member in BFT Group executes the request locally and sends their response back to the cloud module. As soon as the response is collected from the group within a particular period of time the cloud module will judge the consistency of response.

If BFT Group responds consistently the current request will be committed and cloud module will send next request. If it is inconsistent the cloud module will trigger the fault tolerance procedure to tolerate up to f faulty nodes and trigger primary updates/ replica updating procedure to update member. If the fault is more than f then cloud module will resend request to refresh BFT Group and enter into request execution phase again.

(D) PRIMARY UPDATING:

Faulty primary will be identified and be replaced by the newly selected primary

(E) REPLICA UPDATING:

Faulty replicas are identified and updated according to the information acquired from the request execution phase.

III. CONCLUSION:

Fault tolerance plays a major role in the cloud environments to guarantee availability of critical services, application execution and hardware. There is an increased scale and complexity in the cloud computing it is important to ensure the stability, availability and reliability in those systems.

REFERENCES:

[1] 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

[2]Malik, S & Huet, F 2011, 'Adaptive Fault Tolerance in Real Time Cloud Computing', In IEEE World Congress on Services (SERVICES)

[3] Sun, D, Chang, G, Miao, C, & Wang, X 2013, Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments, The Journal of Supercomputing.

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

[5]Tchana, A., Broto, L., & Hagimont, D. (2012, March). Fault Tolerant Approaches in Cloud Computing Infrastructures. In ICAS 2012, the Eighth International Conference on Autonomic and Autonomous Systems

[6] Zhang, Y, Zheng, Z, & Lyu, M R 2011, 'BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing', In IEEE International Conference on Cloud Computing (CLOUD)