www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Vandana Publications
IJEMR

Volume-7, Issue-3, May-June 2017
International Journal of Engineering and Management Research
Page Number: 69-71

Review on Fault Tolerance Techniques in Cloud Computing

SatjotKaur[1], Gurbhej Singh[2]
[1]Department of Computer Science Engineering (USET) Rayat Bahra University, Mohali, INDIA
[2]Assistant Professor, Department of Computer Science Engineering (USET) Rayat Bahra University, Mohali, INDIA

ABSTRACT

Cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time. The dynamic environment of cloud results in various unexpected faults and failures. 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. In this paper, we give a survey on various types, techniques, metrics, challenges and limitations of fault tolerance techniques. It deals with the understanding of fault tolerance techniques and strategies in cloud environment.

Keywords-- HA Proxy, VMs, Cloud Computing, Hadoop, Amazon EC2

I. INTRODUCTION

With the immense growth of internet and its users, cloud computing, with its incredible possibilities in ease, quality of service and on-interest administrations has turned into a guaranteeing figuring stage for both business and non-business computation customers.[4]Cloud computing is a style of computing where service is provided across the Internet using different models and layers of abstraction. It refers to the applications delivered as services 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 centres. A simple example of cloud computing service is Yahoo email or Gmail etc. Due to the exponential and fast growth of cloud computing, the need for fault tolerance in the cloud is a key factor for consideration.

Fault tolerance is one of the key issues in cloud computing. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. Fault tolerance enables a system to continue perform operation, possibly at a reduced level, rather than failing completely, when some subcomponent of the system malfunctions unexpectedly. Fault tolerance is an important aspect in cloud storage, due to the strength of the stored data.[2] The main benefits of implementing fault tolerance in cloud computing include failure recovery, the simple hardware platform they need, being independent from application, high in reliability, lower cost, improved performance metrics etc.

II. TYPES OF FAULTS

These faults can be classified on several factors such as[3]:

A. Network fault: A Fault occur in a network due to network partition, Packet Loss, Packet corruption, destination failure, link failure, etc.

B. Physical fault: This Fault can occur in hardware like fault in CPUs, fault in memory, fault in storage, etc.

C. Media fault: Fault occurs due to media head crashes.

D. Processor fault: Fault occurs in processor due to operating system crashes, etc.

E. Process fault: A fault which occurs due to shortage of resource, software bugs, etc.

F. Service expiry fault: The service time of a resource may expire while application is using it.

III. CATEGORIES OF FAULT

A fault can be categorized on the basis of computing resources and time. A failure occurs during computation on system resources can be classified as: omission failure, timing failure, response failure, and crash failure. Fault may be[3]:

A. Permanent: These failures occur by accidentally cutting a wire, power breakdowns and so on. It is easy to reproduce these failures. These failures can cause major disruptions and some part of the system may not be functioning as desired.

B. Intermittent: These are the failures appears occasionally. Mostly these failures are ignored while testing the system and only appear when the system goes into operation. Therefore, it is hard to predict the extent of damage these failures can bring to the system.

C. Transient: These failures are caused by some inherent fault in the system. However, these failures are corrected by retrying roll back the system to previous state such as restarting software or resending a message. These failures are very common in computer systems

IV. TECHNIQUES FOR FAULT TOLERANCE

A. Reactive Fault Tolerance: Reactive fault tolerance policies reduce the effect of failures on application execution when the failure effectively occurs. Various techniques based on this policy are [1][3]:

1) Check pointing/Restart*:* When a task fails, it is allowed to be restarted from recently checked pointed state rather than from the beginning. In this scenario after doing every change It is an efficient task level fault tolerance for long running and big applications .In this scenario after doing every change a check pointing is done.

2) Replication: Replication means copy. Various tasks are replicated and they are run on different resources, for the successful execution and for getting the desired result. It can be implemented using tools like Hadoop and AmazonEc2 etc. In order to make the execution succeed, various replicas of task are run on different resources until the whole replicated task is not crashed. HAProxy (High Availability Proxy), Hadoop and AmazonEc2 are used for implementing replication. Using tools like HA-Proxy, Hadoop and AmazonEc2 replication can be implemented.

3) Job Migration: On the occurrence of failure, the job is migrated to a new machine. HA Proxy can be used for migrating the jobs to other machines. 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.

4) S Guard: It is less turbulent to normal stream processing. S-Guard is based on rollback recovery. S Guard can be executed in HADOOP, Amazon EC2 (Elastic Cloud Compute).

5) Retry: It is the simplest technique that retries the failed task on the same resource. The user resubmits the task on the same cloud resource. In this case we implement a task again and again.

6) Task Resubmission: A job may fail now. In this case at runtime the task is resubmitted again either to the same machine on which it was operating or to some other machine.

7) User defined exception handling: Here the user defines the specific action of a task failure for workflows.

8) Rescue workflow: It allows the system to keep functioning after failure of any task until it will not be able to proceed without rectifying the fault.

B. Proactive Fault Tolerance: It refers to avoiding failures, errors and faults by predicting them in advance and replace 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 [4]:

1) Self-healing: When multiple instances of an application are running on multiple virtual machines, it automatically handles failure of application instances. A big task can divided into parts .This Multiplication is done for better performance. When various instances of an application are running on various virtual machines, it automatically handles failure of application instances.

2) Pre-emptive Migration: In this technique an application is constantly observed and analysed. Pre-emptive Migration relies on a feedback-loop control mechanism where application is constantly monitored and analysed.

3) Software Rejuvenation: The system is planned for periodic reboots and every time the system starts with a new state.

V. METRICS FOR FAULT TOLERANCE IN CLOUD COMPUTIMG

The existing fault tolerance technique in cloud computing considers various parameters: throughput, response-time, scalability, performance, availability, usability, reliability, security and associated over-head.[1]

A. Throughput: It defines the number of tasks whose execution has been completed. Throughput of a system should be high.

B. Response Time: Time taken by an algorithm to respond and its value should be made minimized.

C. Scalability: Number of nodes in a system does not affect the fault tolerance capacity of the algorithm.

D. Performance: This parameter checks the effectiveness of the system. Performance of the system has to be enhanced at a sensible cost e.g. by allowing acceptable delays the response time can be reduced.

E. Availability: Availability of a system is directly proportional to its reliability. It is the possibility that an item is functioning at a given instance of time under defined circumstances.

F. Usability: The extent to which a product can be used by a user to achieve goals with effectiveness, efficiency, and satisfaction.

G. Reliability: This aspect aims to give correct or acceptable result within a time bounded environment.

H. Overhead Associated: It is the overhead associated while implementing an algorithm. Overheads can be imposed because of task movements, inter process or inter-processor communication. For the efficiency of fault tolerance technique the overheads should be minimized.

I. Cost effectiveness: Here the cost is only defined as a monitorial cost.

VI. LIMITATIONS OF FAULT TOLERANCE

The cloud computing environment still cannot deliver the quality, robustness and reliability that are needed for the execution of various workflows. Providing fault tolerance requires careful consideration and analysis because of their complexity, inter-dependability. Some are below[6]:

- Fault tolerance technique doesn't work when multiple instance are running on different VMs(Virtual Machines).
- All different technologies from vendors of cloud environment need to be unified/ integrated to make dependable system.
- There should be some new technique/ approach to integrate fault tolerance algorithms with existing ones.
- For evaluating the performances of fault tolerance component in comparison with similar ones, a benchmark based method should be developed to ensure high reliability and availability.

VII. CHALLENGES OF FAULT TOLERANCE

As per as the research gaps analysed there is a potential need for implementing autonomic fault tolerance by using different parameters in cloud environment. During the literature review the various challenges faced in incorporating fault tolerance in cloud computing is as follows[2][5]:

- The heterogeneity of the cloud is the biggest hindrance to localize the faults. There is need to implement automatic fault tolerance technique for multiple instances of an application running on several virtual machines.
- There are different technologies from competing various vendors of cloud infrastructure need to be integrated for establishing a reliable system.
- Limited information is provided to the users because of high system complexity, so it is difficult to design an optimal fault tolerance solution.
- Automatic fault tolerance must react to synchronisation among various clouds.
- There are more chances of errors because processing is done on remote computers.
- To ensure high reliability and availability multiple cloud computing providers with independent software stacks should be used
- A benchmark based method can be developed in cloud environment in comparison with similar ones.
- It is difficult to interpret the changing system state because cloud environment are dynamically scalable, unexpected and often virtualized resources are provided as a service.

VIII. SCOPE OF STUDY

Cloud environment is dynamic which leads to unexpected system behaviour resulting in faults and failures. In order to improve reliability and achieve robustness in cloud computing, failures should be assessed and handled effectively. Fault detection is one of the biggest challenges in making a system fault tolerant. Tolerance methods work when a fault enters the boundary of a system. Therefore, in theory, fault-tolerance methods are used to predict the fault and perform an appropriate action, before the faults actually occur. The faults are first detected and then suitable fault tolerance technique (pre-emptive migration/ check-pointing) 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. In the present scenario, there are tolerance error models that introduce different fault-tolerance mechanisms to improve the system. However, there are still challenges that need to be considered for any framework or model. There are weaknesses that cannot complete all aspects of the faults. Thus, it is possible to overcome the weaknesses of all previous models.

REFERENCES

[1] Seyyed Mansur Hosseini and MostafaGhobaeiArani, *Fault-Tolerance Techniques in Cloud Storage: A Survey*, International Journal of Database Theory and Application.

[2] AnjuBala, InderveerChana, Fault Tolerance-*Challenges, Techniques and Implementation in Cloud Computing*, IJCSI International Journal of Computer Science Issues,January 2012.

[3] Dr. Lakshmi Prasad Saikia, *Fault Tolerance Techniques and algorithms in cloud computing*, International Journal of Computer Science &Communication Networks.

[4] Zeeshan Amin, NishaSethi, Harshpreet Singh, *Review on Fault Tolerance Techniques in Cloud Computing*, International Journal of Computer Applications (0975 – 8887) ,April 2015

[5] Pawan Thakur RoohiAli,"Cloud Computing Architecture" in *Cloud Computing 1st Edition New Delhi India TechIndia Publication Series 2013-14*

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