A Survey On Fault Tolerance Techniques And Methods In Cloud Computing

Gagandeep Singh & Supriya Kinger
[1]*Research Scholar (SGGSWU),* [2]*Assistant Professor (SGGSWU)*

ABSTRACT

Cloud computing is the next generation computing. The GUI which controls the cloud computing makes is directly controlling the hardware resource and application. This paper based on a survey of different kind of faults and their prevention methods. There are several methods used to avoiding faults in real time distributed systems and cloud computing, that survey is devoted to study of introductory concepts and techniques for designing and analyzing fault tolerant systems. Fault tolerant system is one that can provide continue correct performance of its specified tasks in the presence of hardware and/or software faults.

General Terms

Fault tolerance, Dependability, Redundancy, and Load balancing

1. Introduction

Cloud computing is a comprehensive solution to delivers IT as a service. It is an internet based computing solution where shared resources are provided as electricity. Cloud computing is a way of computing where service is provided across the internet using models and levels of abstraction [1]. The flexibility of cloud computing is a function of allocating resource on demand. Cloud computing is the combination of grid computing and utility computing [2]. Many research issues are fully addressed in cloud such as Fault tolerance, security, etc. Fault tolerance is an important key issue 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 etc. [3]. 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.

2. Background Terminology

A fault tolerance is a setup or configuration that prevents a computer or network device from failing in the event of an unexpected problem or error [4]. To make a computer or network fault tolerant requires that the user or company to think how a computer or network device may fail and take steps that help prevent that type of failure. The path of generation of fault is shown in a figure 1.

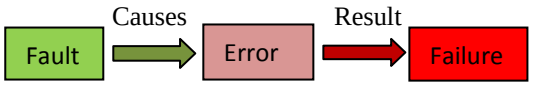

Figure 1. Generation path of failure.

- A system is said to fail when it will not fulfill the requirements.
- An error is part of the system state that may lead to a failure.
- The cause of an error is a fault [5].Transient, intermittent or permanent faults, Design faults or operational faults.

3. Dependability and Classification

Fault tolerance makes to achieve system dependability. Dependability is related to some QoS aspects provided by the system, it includes the attributes like reliability, safety and availability [6]. The dependability classified as shown in figure 2. Fault avoidance is preventing faults from occurring, focus on methodologies for design, testing, and validation. Fault tolerance is use protective redundancy to mask failures, focus on how to use components in a manner such that failures can be masked.

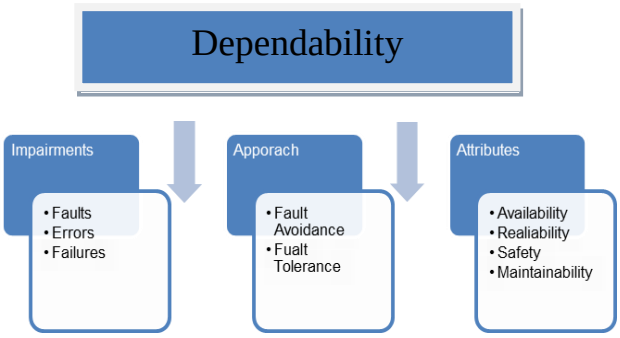

Figure 2. Dependability with classifications.

4. Fault Tolerance Techniques

Fault tolerance techniques is use to overcome the faults so that system run smoothly throughout the process. There are number of methods to deal with faults which are as a redundant method, recovery methods, load balancing.

4.1 Hardware Redundancy

Hardware redundancy is a structural redundancy technique that completely masks faults within a set of redundant modules. A number of identical modules execute the same functions, and their outputs are voted to remove errors created by a faulty module [7]. Triple modular redundancy (TMR) is a commonly used form of fault masking in which the circuitry is triplicated and voted as shown in figure [3].

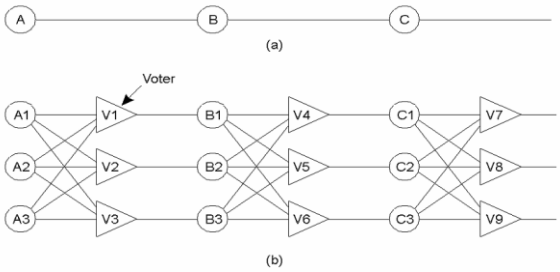

Figure 3. Triple modular redundancy.

The voting circuitry can also be triplicated so that individual voter failures can also be corrected by the voting process. A TMR system fails whenever two modules in a redundant triplet create errors so that the vote is no longer valid. Hybrid redundancy is an extension of TMR in which the triplicated modules are backed up with additional spares, which are used to replace faulty modules allowing more faults to be tolerated.

4.2 Software Redundancy

In software redundancy method uses two different programs or algorithms, beyond what is needed to perform a given function, to detect and possibly tolerate faults. It independently produces two or more versions of that software in hope that the different versions will not fail on the same input. These forms of design diversity will ensure that not all copies will fail on the same set of input data [8].

4.3 Software Redundancy

In time redundancy a technique is recomputed or redoes the task and compares the results, it may or may not be use the same hardware or software. It is uses additional time to performing the functions of a system such that fault detection and often fault tolerance can be achieved. The systems exploit time redundancy through re-execution of the same program on the same hardware [8].

4.4 Fault Tolerance Based on Policies

Fault tolerance (FT) policies can typically be categorized into two sets: reactive fault tolerance policies and proactive fault tolerance policies. While reactive fault tolerance policies minimize the impact of failures on application execution when the failure effectively occurs; proactive fault tolerance policies aim at predicting failures and move running applications away from nodes that are predicted to fail [9].

4.5 Load Balancing Fault Tolerance

Load balancing is basically working on traffic shaping it may handles the inbound or outbound traffic in any network and load balancing further divided into sub categories.

4.5.1 Hardware Based Load Balancing

Hardware based load balancing directs client requests for a single IP address to multiple hosts within a cluster. Hardware load balancers typically use a technique called network address translation (NAT), which exposes one or more virtual IP address to clients and forwards data for the designated hosts by translating IP addresses and resending network packets. This technique introduces a single point of failure, the computer performing the redirection of packets, between the cluster and the clients. To achieve high availability with this solution, you need a backup load balancer [10].

4.5.2 Dispatcher Software Load Balancing

This load balancing solution requires one dispatch server to handle all incoming connection requests, where they are then retransmitted to other servers in the network. This solution limits throughput and restricts performance because the entire cluster's throughput is limited by the speed and processing power of the dispatch server. The single dispatch server represents a single point of failure, which must be eliminated by moving the dispatching function to a second computer after a failure occurs [10].

4.5.3 Network Load Balancing

Network load balancing is a fully distributed, software based solution and does not require any

specialized hardware or network components. Network load balancing is not requiring a centralized dispatcher because all hosts receive inbound packets, and redundancy is provided according to the number of hosts within the cluster. The filtering algorithm for network load balancing is much more efficient in its packet handling than centralized load balancing programs, which must modify and retransmit packets [10].

5. Conclusion and Future scope

This paper we described types of faults, dependability taxonomy, and various fault tolerance techniques, and broadly classified into categories. We described the some methods which are preventing the faults which are redundancy methods, fault tolerance policies, load balancing fault tolerance. Our future work will be based on drawback of fault tolerance policies, we purposed new framework which is based on virtualization and we will follow a reactive fault tolerance approach with network load balancing fault tolerance technique helps to eliminate the faults.

6. REFERENCES

[1] L. Arockiam, S. Monikandan& G. Parthasarathy," Cloud Computing: A Survey, International Journal of Internet Computing (IJIC), and ISSN No: 2231 – 6965, Volume-1, Issue-2, 2011.

[2] Qi Zhang, Lu Cheng,RaoufBoutaba," Cloud computing: state-of-the-art and research challenges, J Internet ServAppl (2010) 1: 7–18 DOI 10.1007/s13174-010-0007-6.

[3] Y.M. Teo, B.L. Luong, Y. Song, T. Nam,"Cost-Performance of Fault Tolerance in Cloud Computing, International Conference on Advanced Computing and Applications, (Special Issue of Journal of Science and Technology, Vol. 49(4A), pp. 61-73), Ho Chi Minh, Vietnam, October 19-21, 2011.

[4] Ravi Jhawar, Vincenzo Piuri, Marco Santambrogioy," A Comprehensive Conceptual System-Level approach to Fault Tolerance in Cloud Computing,DOI 10.1109/SysCon.2012.6189503.

[5] A. Christy Persya,Sr.Lecturer, T.R.Gopalakrishnan Nair," Fault tolerant real time systems, International Conference on Managing Next Generation Software Application (MNGSA-08), Coimbatore, 2008.

[6] AlgirdasAvizienis, Jean-Claude Laprie, Brian Randell," Dependability and its Threats: taxonomy.

[7] Avizienis, (Ed), "Dependable Computing and Fault-Tolerant Systems Vol. 1", The Evolution of Fault-Tolerant Computing, Vienna: Springer-Verlag.

[8] TimothyTsai," Fault tolerance via N-Modular Software Redundancy, FTCS`98 Munich Germany, 23-25,06,1998.

[9] KulathepCharoenpornwattana, GeoffroyVallee, Christian Engelmann, AnandTikotekar, Thomas Naughton, Stephen L. Scott, ChokchaiLeangsuksun, "A Framework for Proactive Fault Tolerance", ARES 2008-Barcelona, Spain.

Network load balancing: http://msdn.microsoft.com/enus/library/windows/desktop/cc296105%28v=vs.85%29.aspx