

Proactive Faults Tolerance Review for Increasing the Reliability of Virtual Machines in Cloud Computing

Dr Suchithra Suriya
Professor & Head,
Dept of Masters of Sc. in IT

Uwayo Serge Gaston
Msc. IT, Student,
Dept of Masters of Sc. In IT

Abstract-- Fault tolerance is a main concern to guarantee reliability of services as well as application execution of the virtual machines. In order to minimize failure impact of the virtual machine and application execution, failures should be anticipated and proactively handled. Fault tolerance techniques are used to predict these failures and take an appropriate action before failures actually occur. This paper aims to provide a better understanding of proactive fault tolerance challenges and identifies various tools and techniques used for fault tolerance.

Keywords: Cloud computing, virtual machines, Proactive.

1. INTRODUCTION:

Cloud computing is an evolving large -scale computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.[1]

Cloud computing is built upon virtualization, distributed computing, utility computing, and more recently networking, web and software services. Individuals and organizations use hardware and software managed by third parties at remote location. Online file Broad network access: Accessing resources is available over the network and accessed through standard mechanisms that promote use of the most common devices (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often

storage, social networking sites, webmail, and online business applications are some common cloud services. User can use these services without knowing the underlying hardware and software details.

Cloud computing has been widely adopted by the industry, still there are many research issues to be fully addressed like fault tolerance, workflow scheduling, workflow management, security etc. Fault tolerance is one of the key issues amongst all. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. When a fault occurs, these techniques provide mechanisms to the software system to prevent system failure occurrence [18]. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics etc.

Important characteristics:

On-demand self-service: A customer is given resources on demand by the cloud computing. Instead of consulting the IT interaction, the user performs all the action needed to complete the service himself, this is made possible by capability of self-service and automation.

appear to be unlimited and can be appropriated in any quantity at any time.

Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. [2]

Fault tolerance is necessary for the system to guarantee both availability and reliability of critical services and application execution. To minimize the occurrence of failures in the system and its impact on application execution, failures should be handled by suitable technique. The aim of this survey is to analyze the existing proactive fault tolerance techniques in cloud computing in order to overcome the shortcoming of the existing techniques and

developing new algorithm for predicting fault in appropriate way.

2. SERVICE MODELS:

Software as a Service (SaaS): Applications are hosted by the service provider to the consumer.

The accessibility of the application are made possible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).[2]

Hardware as a Service (HaaS): HaaS permit the users to access the whole computing power . The users choose the operating system on the back end, as well as the number of virtual Machines (VMs) running on the hardware.

3. PROACTIVE FAULT TOLERANCE.

The principle of Proactive fault tolerance polices is to anticipate the fault and avoid recovery from fault, errors and failures proactively replace the suspected node with other node (i.e.) it detects the problem before it actually comes. It uses an anticipation mechanism that tolerates faults by relying on the system log as well as the hardware or software state. It prevents compute node failures from impacting running parallel applications by preemptively migrating parts of an application (task, process, or virtual machine) away from nodes that are about to fail. Some of the techniques based on these policies are preemptive

Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS): The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. migration, Software Rejuvenation and using self-healing [5].

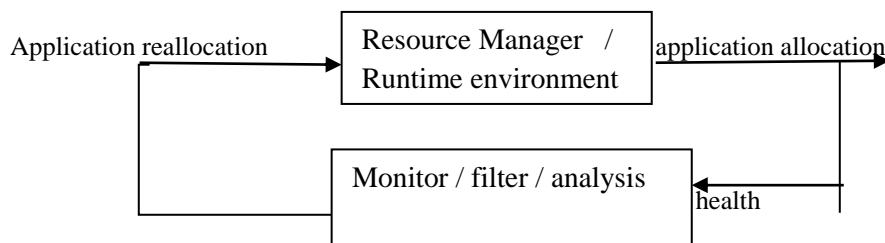
3.1 PREEMPTIVE MIGRATION:

It is a concept that prevents compute node failures from impacting running parallel applications by preemptively migrating parts of an application (task, process, or virtual machine) away from nodes that are about to fail. Pre-fault indicators, such as a significant increase in heat, can be used to avoid an imminent failure through anticipation and reconfiguration. Since avoiding a failure through preemptive migration is significantly more efficient than recovery from failure via traditional reactive fault tolerance mechanisms, such as checkpoint/restart, HPC system utilization becomes more efficient.

The preemptive migration count on feedback loop control mechanism where the application is constantly monitored and analyzed.

Proactive Fault Tolerance using preemptive migration relies on a feedback-loop control mechanism (Fig. 1), where the health of each application running on a HPC system is monitored and preventative action is taken to avoid imminent application failure by reallocating running application parts from unhealthy to healthy compute nodes [6]

Fig. 1 Feedback loop control of proactive fault tolerance using Preemptive migration



3.2 SELF HEALING

Table 1: self-healing system’s faults and fixes

Faults	Fixes
Source Code Bugs	Reboot Tier / Service and Notification to Administrator
Java Exceptions	Micro-boot EJB
Deadlock Threads	Micro-boot EJB, Kill hung Query
Buffer Contention	Repartition the memory across various Buffers
Aging	Rebooting for reclaiming the leaked resources
Read / Write Contention on Table Block	Repartition Table to Balance Accesses around Partitions

The Self-healing term is used to mean to recover from run-time failures. A system needs to be able to notice an error or injury and then to act on it. A big task will be divided into parts.

Software, which is able to detect and react on its malfunctions, is called self healing software.

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.

Failure of an instance of an application running on multiple virtual machines is controlled automatically, that is the capability of a software system to deal with bugs will be done automatically.

Fault tolerance for dependable computing is to provide the specified service through rigorous design whereas self-healing is meant for run-time issues.

A fault-model of Self-Healing system is to state what faults or injuries to be self-healed containing fault duration, fault source such as, operational errors, defective system requirements or implementation errors etc. [8]

4. FAULT TOLERANCE MODELS

AFTRC: a fault-tolerance model for real time cloud computing based on the fact that a real time system can take advantage the computing capacity, and scalable virtualized environment of cloud computing for better implement of real time application. In this proposed model the system tolerates the fault proactively and makes the diction on the basis of reliability of the processing nodes

Adaptive fault tolerance in real time cloud computing. The system tolerates the fault and makes the decisions on the basis of reliability of the processing nodes named virtual machines. After every computing cycle the reliability of virtual machines changes that’s why it is adaptive. But if a virtual machine manages to produce a correct result within

the time limit than the reliability increases but if it fails then the reliability of virtual machine decreases. [9]

LLFT: is a propose model which contains a low latency fault-tolerance (LLFT) middleware for providing fault-tolerance for distributed applications deployed within the cloud computing environment as a service offered by the owners of the cloud. This model is based on the fact that one of the main challenges of cloud computing is to ensure that the application which are running on the cloud without a hiatus in the service they provided to the user. This middleware replicates application by the using of semi-active replication or semi-passive replication process to protect the application against various types of faults [10].

FTWS: is a proposed model which contains a fault-tolerant work flow scheduling algorithm for providing fault-tolerance by using replication and resubmission of tasks based on the priority of the tasks in a heuristic metric. This model is based on the fact that work flow is a set of tasks processed in some order based on data and control dependency. Scheduling the workflow included with the task failure consideration in a cloud environment is very challenging. FTWS replicates and schedule the tasks to meet the deadline [11].

FTM: is a proposed model to overcome the limitation of existing methodologies of the on-demand service. To achieve the reliability and resilience they propose an innovative perspective on creating and managing fault-tolerance. By this particular methodology user can specify and apply the desire level of fault-tolerance without requiring any knowledge about its implementation. FTM architecture this can primarily be viewed as an assemblage of several web services components, each with a specific functionality [12].

CANDY: is a component base availability modeling frame work, which constructs a comprehensive availability model semi automatically from system specification describe by

systems modeling language. This model is based on the fact that high availability assurance of cloud service is one of the main characteristic of cloud service and also one of the main critical and challenging issues for cloud service provider [13].

Vega-warden: is a uniform user management system which supplies a global user space for different virtual infrastructure and application services in cloud computing environment. This model is constructed for virtual cluster base cloud computing environment to overcome the problems: usability and security arise from sharing of infrastructure [14].

FT-Cloud: Are a component ranking based frame work and its architecture for building cloud application. FT-Cloud employs the component invocation structure and frequency for identify the component. There is an algorithm to automatically determine fault tolerance stately [15].

MAGI-CUBE: a high reliable and low redundancy storage architecture for cloud computing. The build the system on the top of HDFS and use it as a storage system for file read /write and metadata management. They also built a file scripting and repair component to work in the back ground independently. This model based on the fact that high reliability and performance and low cost (space) are the 3 conflicting component of storage system. To provide these facilities to a particular model Magi cube is proposed [16].

5. CONCLUSION

The proactive Fault Tolerance has an important role in supporting the organized cloud environment. When the faults enter into the system then the fault tolerance methods is needed.

This survey paper reviewed the various proactive Fault Tolerance and fault tolerance models.

But none of them can fulfill all aspects of faults tolerance. Thus in future we review the reactive faults tolerance in order to try to maximize fault tolerance aspect in cloud infrastructure.

6. REFERENCES

- [1] Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). Ieee
- [2] Mell, P., & Grance, T. (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6), 50.
- [3] B. P. Rimal, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," 2009, pp. 44–51.
- [4] Engelmann, A. Geist and C., "Development of naturally fault tolerant algorithms for computing on 100,000 processors," 2002. <http://www.cs.ucla.edu/~rennells/article98.pdf>
- [5] C. Engelmann, G. R. Vallee, T. Naughton and S. L. Scott, "Proactive Fault Tolerance Using Preemptive Migration," *2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Weimar, 2009, pp. 252-257.
- [6] <http://www.cloudbus.org/papers/SoftwareRejuvenationCloud2015.pdf>
- [7] <http://www.clei.org/cleiej/papers/v10i2p5.pdf>
- [8] Sun Microsystems, Inc., "Introduction to Cloud Computing Architecture", White Paper 1st Edition, (2009).
- [9] W. B. Zhao, P. M. Melliar and L. E. Mose, "Fault Tolerance Middleware for Cloud Computing", *IEEE 3rd International Conference on Cloud Computing*, (2010).
- [10] https://books.google.co.in/books?id=m09zpGjUPIAC&pg=PA666&lpg=PA666&dq=FTWS+fault+tolerance&source=bl&ots=x8AcPSzVPJ&sig=dibBUpt1dQWXB C2HYhX3kO231dg&hl=en&sa=X&ved=0ahUKEwj5vqN_4_MAhVEJqYKHfkRDe0Q6AEIUTA#v=onepage&q=FTWS%20fault%20tolerance&f=false
- [11] R. Jhwar, V. Piuri and M. Santambrogio, "A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing", *IEEE*.
- [12] F. Machida, E. Andrade, D. S. Kim and K. S. Trivedi, "Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML", *30th IEEE International Symposium on Reliable Distributed Systems*, (2011).
- [13] J. Lin, X. Y. Lu, L. Yu, Y. Q. Zou and L. Zha, "Vega Warden: A Uniform User Management System for Cloud Applications ", *Fifth IEEE International Conference on Networking, Architecture, and Storage*, (2010).
- [14] Z. B. Zheng, T. C. Zhou, M. R. Lyu and I. King, "FT-Cloud: A Component Ranking Framework for Fault-Tolerant Cloud Applications", *IEEE 21st International Symposium on Software Reliability Engineering*, (2010).
- [15] Q. Q. Feng, J. Z. Han, Y. Gao and D. Meng, "Magi-cube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing", *IEEE Seventh International Conference on Networking, Architecture, and Storage*, (2012).