



# Proactive Fault Tolerance Policy for Virtual Machine Management in Private Cloud Environment

Sudha M, Usha J

**Abstract:** Cloud computing being a new delivery model has many research challenges. It opens the door to great amount of innovative methods and practices. Many researchers have come-up with novel and efficient fault tolerant solutions for the cloud. With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model. Most of the research on fault tolerant algorithms, mechanisms and techniques are focused on datacenters. In this paper we propose fault detection and prevention policies for VM creation lifecycle derived from cloud computing patterns.

**Keywords :** Fault Tolerance, Proactive policies, patterns, Virtual Machine.

## I. INTRODUCTION

### A. Cloud Computing

Cloud computing being a new delivery model has many research challenges. It opens the door to great amount of innovative methods and practices. Many researchers have come-up with novel and efficient fault tolerant solutions for the cloud. With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model. Most of the research on fault tolerant algorithms, mechanisms and techniques are focused on data-centre based environments. Since accessibility to these kind of infrastructure is limited, they are experimented on simulators/simulated environment. The key technology

player of cloud computing is the virtualization technology. The physical resources are emulated as virtual resources giving the user a feel of the actual hardware resource. A Virtual Machine (VM) is efficient duplication of the physical machine. A hypervisor is a software that provides virtual resources. It acts as a transparent layer between the physical resources and the virtualized resources. Two types is hypervisors are Type I and Type II. In Type I hypervisors a guest OS resides above the hypervisor, example XEN. In type II The hypervisor runs above the guest operating system, example KVM. Virtual Machines emulate the behavior of the underlying hardware. There are different types of virtualization approaches, LXC, containers, full virtualization, para virtualization. Comparison of these approaches are discussed in [1,2].

### B. Fault Tolerance

Fault tolerance is the capability of a system to perform even in the presence of failures. To build a complete fault tolerant system is a myth, but building it to be resilient to failures need to be achieved. This work aims at designing policies to build a fault tolerant private cloud deployment by defining policies to detect and prevent possible failures. The policies are defined based on well-known cloud computing patterns. Cloud services are on-demand and pay-per-use model and hence must provide highly available and reliable environment. Cloud services are accessed over heterogeneous devices. Many cloud service providers build and maintain huge datacenter farms and provision the resources on demand to the users. Great amount of effort and money has been put in to build and maintain these huge datacenter farms which are highly available, fault tolerant, reliable and secure. To ensure high availability, datacenters are replicated. Replication is a very expensive but a very reliable solution. Faults, Errors and Failures: Software or hardware defects leads to errors and some may escalate and leads to critical failures. If initial failures are not addressed it cascades with further failures. These failures may be trivial like transmission of single IP packet arriving out of order or being corrupted or it could be a serious application failure that could breakdown the entire system. The primary characteristics of failure a service impact is the duration of service disruption.

Revised Manuscript Received on December 30, 2019.

\* Correspondence Author

**Sudha M\***, Department of MCA, RVCE-Research Centre, Visvesvaraya Technological University, Bangalore, India,

**Usha J**, Department of MCA, R V College of Engineering, Visvesvaraya Technological University, Bangalore, India,

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Fault->Fault Activation -> Error-->Failure-> Breakdown

Fault tolerance is the ability of a system to perform even in the presence of failures. To build a complete fault tolerant system is a myth, but building it to be resilient to failures need to be achieved. This work aims at designing policies to build a fault tolerant private cloud deployment by defining policies to detect and prevent possible failures. The policies are defined based on well-known cloud computing patterns. Cloud services are on-demand and pay-per-use model and hence must provide highly available and reliable environment. Cloud services are accessed over heterogeneous devices. Many cloud service providers build and maintain huge datacenter farms and provision the resources on demand to the users. Great amount of effort and money has been put in to build and maintain these huge datacenter farms which are highly available, fault tolerant, reliable and secure. To ensure high availability, datacenters are replicated. Replication is a very expensive but a very reliable solution.

## II. BACK GROUND WORK

Faults tolerance techniques are classified as reactive and proactive techniques based on their policies. While Reactive Fault Tolerance policies focus on reducing the impact of failure on applications when the failure actually occurs, the proactive FT policies aims at prevention. These policies focus on predicting and preventing the faults, failures and errors even before they occur and proactively identify and replace the error causing component. The fault tolerant techniques are discussed in this section :Vega-warden, a standardized user management system which provide a universal user space for different virtual infrastructure and application services. The model is built on virtual cluster based cloud computing environment to prevail over the issues like usability and security resulting from resource sharing [3]. FT-Cloud is a framework based on component ranking and design for building cloud apps. The framework uses component invocation structure and frequency for identify the component. There is an algorithm to automatically determine fault tolerance stately [4] Magi-Cube [5], reliable and low redundancy storage architecture is built on the top of HDFS and uses it as a storage system for file read /write and metadata management. The model is a high reliable, and performance and low space storage system. With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model. With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and

demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model. Parameters to measure faults are Mean Time between Failure (MTBF), Availability, Failure rate, Latency, Throughput, Threshold, Memory Usage, CPU Load, CPU Temperature, and Time Taken with respect to VM creation. The policies are derived from cloud computing patterns. Pattern gives a proven solution to a common problem, which is documented individually documented in consistent format. [6] It is generally part of a bigger collection. In the IT scenario patterns is centered around the designing of automated systems and are termed as *design patterns*. CC patterns can be classified into two categories one that characterize cloud providers and other describe guidelines for building cloud applications. [7]

Pattern Category	Pattern Subcategory
Cloud Provider related Patterns	Cloud Environment
	Cloud Service models
	Cloud offerings
Cloud Application related Patterns	Cloud Application Architecture
	Application Management
	Composite Applications

With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model. With the rapid advancement in internet technology, usage of smart devices and social networking are giving rise to tremendous amount of data and demands resources like never before. Cloud computing delivery model provides solution to the need of the day. The recent advancement in IoT, Edge computing etc, has expanded the scope of visualizing cloud computing with a different perspective. There is a need for re-provisioning the existing components of the cloud model.

## III. METHODOLOGY

This chapter focuses on the Lower Layer of Cloud Computing Services, Infrastructure as a Service (IaaS). The cloud infrastructure consists of pool of resources that are provisioned to the user on demand. Resources such as compute, storage and network are combined into a large pool of virtual resources. The virtual resources are further provisioned to the uses. Virtualization is the key technology player in cloud infrastructure services. A hypervisor is a software that emulates the functionalities of the underlying hardware. The faults in the IaaS layer can escalate to the above layers over a period of time leading to catastrophic failure of the system.

Therefore it is very crucial for the cloud services to monitor the resources and proactively handle the failures. In this paper we propose the proactive fault tolerance policies to detect faults and prevent failure.

In this section private cloud environment using commodity hardware has been explained. The main objective of the work is to build a Fault Tolerant cloud stack for private cloud environment and proposed fault detection and prevention methods.

**A. Infrastructure as a Service (IaaS) Stack**

An IaaS stack is built that is resilient to failures, by imbuing the fault tolerance capability with in the stack. Thus, the deployed private cloud environment is fault tolerant stack. The fault tolerant layer has two modules fault detection module and fault prevention module which interacts with the application layer on the top and the infrastructure services in the lower level. There are many readily available stack which can be deployed on specialised hardware in a datacenter. But there is hardly any provision for experimenting in IaaS layer in enterprise class cloud stack in such a kind of environment. With the recent advancement in cloud applications, a simpler and robust stack will lead to a greater adoption of this environment. The IaaS stack is designed for low-end commodity hardware which is light in terms of resource consumption but yet a Fault Tolerant stack. The presently available stacks are not targeted to this kind of requirement, and hence not readily adaptable, thus opening up research opportunity. The Fault Tolerant Cloud Model for low-end commodity infrastructure and develop fault tolerant module which broadly addresses fault detection and fault prevention policies in Infrastructures Services, VM Life cycle and Application Deployment. The conceptual Model of fault tolerant private cloud is shown in Fig. 1.

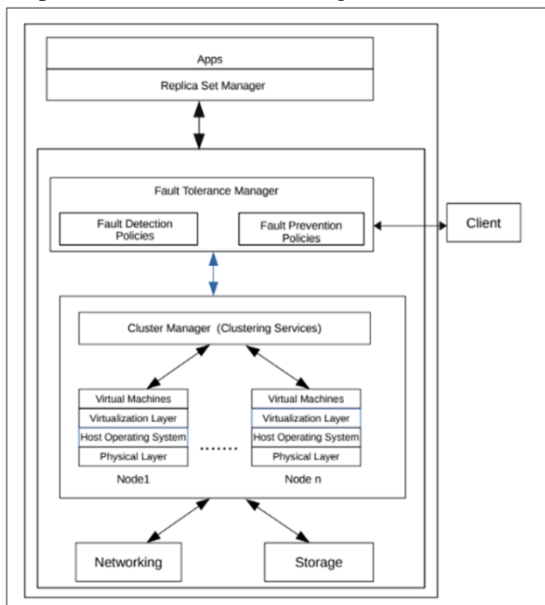


Figure 1. : Fault Tolerant IaaS stack

The IaaS stack is build using open source tools. The tools are choose such a way that they carry the essential functionalities such that the overhead on the infrastructure is reduced. CPU

Load is the key performance indicator. The stack is built to provision compute Instances to the user. Fig. 2 depicts the IaaS stack with the tools used.

<b>Users</b> (Browser, Mobile Device, Apps)
<b>Fault Tolerant Policies</b> (Fault Detection and Prevention)
<b>Clustering</b> (CloudLand/Ansibel)
<b>Hypervisor</b> (FireCracker/Ignite)
<b>Operating System</b> (Debian/KVM/Tap Devices)
<b>Hardware</b> (Physical Machine)

Figure 2 : IaaS Stack Implementation

**B. Fault Detection and Prevention Policies**

Proposed policies related to the VM creation lifecycle are defined here which enforces guidelines for managing a VM creation life cycle. Assumptions made are, it is a private cloud deployment, self hosted and single tenancy model. It is targeted at low-end commodity hardware and an immutable infrastructure. Immutable infrastructure is an approach where in rather than change, components are replaced. We propose policies devised on the assumptions and are derived from cloud computing patterns. Some patterns considered in this paper are resource pooling, watch dog, dynamic failure detection and recovery.

Policies are designed as Assumptions and Design Decision made.

Assumption 1 : Reduce impact on Infrastructure  
Decision : Induce latency between VM Set creation and observe CPU load and CPU Temp. for VM Set creations.

Parameters : Latency, CPU Load, CPU Temperature.

Assumption 2: Need to maintain the stability of the VM  
Decision : To define band or range of suitable VM capacity and VM capacity specified by application

Parameters : band/range

Assumption 3: Detect VM failure during creation process  
Decision: To define the Time to respond after successful creation. (Response Time wrt to VMCLC) and to ensure VM accessibility over the network using IP/MAC address.

Parameter : Response-Time, availability

These policies are implemented on the test-bed and the results obtained are based on the test cases. Two test cases are considered for verification of the proposed policies.

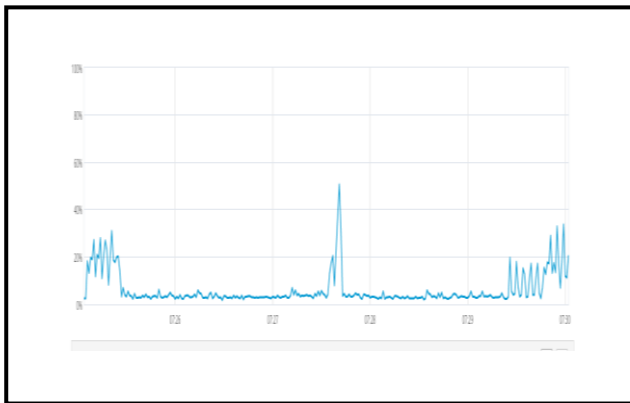
**IV. RESULT AND DISCUSSION**

The results of the two test cases, TC1 and TC2 for creation of MicroVMs were obtained for Five runs for each test respectively. This test was conducted for testing the VM creation life cycle process with CPU performance parameter.

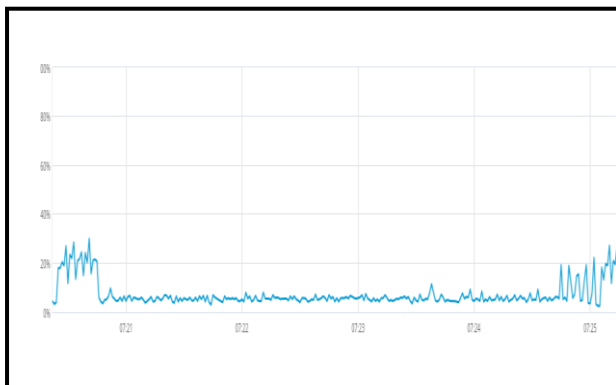
**Table 1 : VM Creation Test Case**

Test Case ID :TC1	
Test Case Scenario : VM Creation with TTL	
Test ID	Test Scenario
TC1.T1	5 VM TTL 5 Mins Runs 3
TC1.T2	5 VM TTL 10 Mins Runs 3

The test cases helped in identifying suitable VM TTL for running different types of jobs. The test was conducted to study the CPU load from the respective job types. It was concluded that Testcase TC1.T1 was suitable from short jobs and TC1.T2 was suitable for jobs with longer run time. The results obtained from the test runs are depicted in Figures 3,4 respectively.



**Figure 3: Testcase TC1.T1 First Run**



**Figure 4. Testcase TC1.T1 Second Run**

## V. CONCLUSION

A fault tolerant IaaS stack for a private cloud environment built on commodity hardware has been proposed in this paper. Virtual Machine related policies are explained. The stack has been discussed and fault detection and prevention policies are derived based on the given assumptions. Tests were conducted to check the impact on the infrastructure during VM creation process. The time-to-live of VM was considered to determine the best case for job scheduling, short running jobs and longer running jobs. It can be concluded that if jobs are scheduled according to the policies defined then failure can be reduced significantly or completely.

## REFERENCES

1. Sudha M, Harish G M, Usha J, "Performance Analysis of Linux Containers - An Alternative Approach to Virtual Machines", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014 ISSN: 2277 128X, pp 820-824
2. Sudha M, Harish G M, Nanadan A, Usha J, "Performance Analysis of Kernel-based Virtual Machines", International Journal of Computer Science & Information Technology (IJCSIT) Vol 5, No 1, February 2013, pg 137-144
3. Jianlin, Xiaoyi Lu, Lin Yu, YongqiangZou and Li Zha "Vega Warden: A Uniform User Management System for Cloud Applications "2010 Fifth IEEE International Conference on Networking, Architecture, and Storage.
4. Zibin Zheng, Tom Chao Zhou, Michel R. Lyu, and Irwin king "FTCloud: A Component Ranking Framework for FaultTolerant Cloud Applications "2010 IEEE 21st International Symposium on Software Reliability Engineering.
5. Qingqing Feng, Jizhong Han, Yun Gao, Dan Meng "Magicube: High Reliability and Low Redundancy Storage Architecture for Cloud Computing" 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage.
6. Thomas Erl, Rober Cope, Amin Naserpour, "Cloud Computing Design Patterns", The Prentice Hall Service Technology Series, 1st Edition, 2015 : ISBN-13:978-0133858563
7. Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter, "Cloud Computing Patterns", Springer, 2014.

## AUTHORS PROFILE



**Sudha M**, Masters in Computer Applications, presently pursuing research at RVCE-MCA research Centre, Visvesvaraya Technological University, Bangalore. Her research area is Cloud Computing. She 24 yrs of work experience of which 5 years of teaching Her area of interest include cloud computing, distributed computing, networking, databases.



**Usha J** is working as Professor, Dept of Master of Computer Applications (MCA) Department, RV. College of Engineering, Bengaluru. She obtained her B.Sc(PCM) and MCA from Bangalore University. She has over Twenty years of experience in research and teaching. She obtained her Ph.D from University of Pune in 2011. She is a Senior member of IEEE and Life member,CSI and ISTE. She has published over 50 papers in International Conferences and Journals.