



Fault Tolerance Policy for Resilient Private Cloud Environment

Sudha M, Usha J

Abstract: Cloud Computing, being a delivery model is swiftly moving ahead by being adopted by small and large organization alike. This new model opens up many research challenges. As, cloud computing services are offered over the Internet on pay-per-use basis, it is very essential to provide fault tolerant services to the users. To ensure high availability, data centers are replicated. The process of replication is costly but in terms reliability it overtakes the cost factors. Vast amount of work has been undertaken in fault tolerance in other computing environments but they cannot be applied directly to the cloud. This gives an opportunity for new, effective solutions. In this paper, we propose policies for delivering fault tolerant services for private cloud computing environment related to virtual machine allocations. The experimental test results and policies derived are described with respect to virtual machine provisioning.

Keywords: Virtualization, Hypervisor, Immutable Infrastructure, private cloud, fault tolerance

I. INTRODUCTION

A. Cloud Computing

Presently, heterogeneity of users, clients and applications, have revolutionised the way Cloud Computing is moving forward. The Cloud Computing delivery model provisions computing resources that resides in datacentres of the Cloud Service Providers, over the Internet adopting Pay-per-use model. With the ever increasing adoption of Cloud, the need for rapid expansion of the size of datacentres is high. Along with the expansion comes the increased risk of resource failures [1] which can degrade the performance of the services provided. Virtualization technology, one of the key enablers of Cloud Computing, is being pushed more than ever [2]. Various implementations of Virtualization layer is being experimented and implemented. Cloud is still in infancy and thus lacks standards and practices [3]. The critical issues that needs to be addressed in the Cloud Computing model include workflow scheduling [4], reliable computing, fault tolerance services, load balancing [5], availability of services[6]. There are many solutions to these existing issues in other computing environments, but applying them directly here is not viable. It needs a new approach and a relook into the existing issues. Fault Tolerance is one of the key factors for reliable computing environment [7,8].

Revised Manuscript Received on December 30, 2019.

* Correspondence Author

Sudha M*, Department of MCA, RV College of Engineering-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 article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Cloud computing business model offers services over the Internet. NIST has defined three specific service models and four deployment models for the Cloud Computing. The lowest level being Infrastructure as a Service (IaaS) where in the compute, storage and networking resources are delivered as Services. The middle layer Platform as a Service (PaaS) delivers development platforms, databases and web servers etc as a service and Software as a Service (SaaS) where software applications are delivered as services. Figure.1 depicts the three specific services offered by the Cloud model. The four deployment models are private, public, hybrid and commodity clouds. Software development process and software architecture optimization is also a challenge that Cloud model needs to standardize[9]

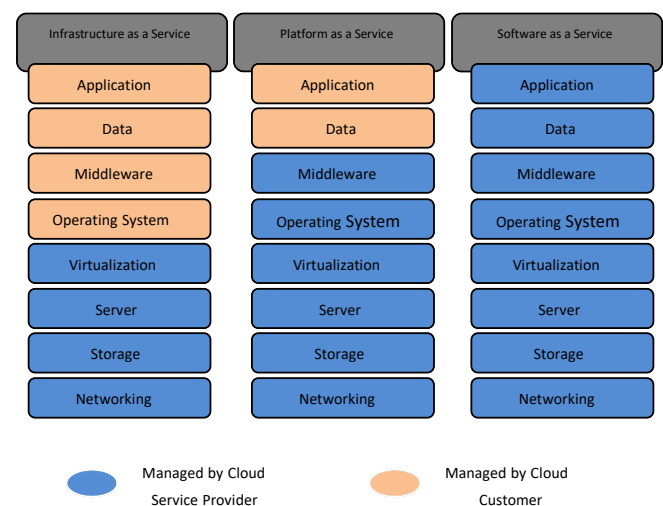


Figure 1: Cloud Service Model

II. BACK GROUND WORK

The ability of a system to continue functioning even in presence of failures is termed as Fault Tolerance. It is an exhaustive field of research, irrespective of the system/area in which it is essential. A good amount of research has been done in this area with many new techniques and algorithms being proposed and implemented along with much new architecture being proposed.

In[10], VM Load Balancing Algorithm “Weighted Active Monitoring Load Balancing Algorithm” is discussed. In this a weight is assigned to all the available VMs in the datacenter to effectively load balance to attain improved performance parameter. Virtual Machines with higher weights are powerful VMs and are assigned the jobs first and then to the lower weight Virtual Machines.

Load-balanced task scheduling algorithm is discussed in [11] with double level task scheduling process providing higher resource utilization. In [12], optimized task scheduling on the basis of algorithm of genetic simulated annealing has been described. The quality of service parameters like, completion-time, bandwidth, price and distance have been taken into consideration. Annealing is applied after the selection, crossover and mutation, to improve local search ability of the genetic algorithm.

Scaling and deploying application has become easier with immutable/irreversible infrastructure. The present day's approach aims at controlling the architecture based on a known VM state and provide VMs with essential dependency [9]. This is accomplished by continuous delivery [13,14]. Authors in [15], explain the significance of micro services and container model for the infrastructure and alleviation of migration with appropriate containerization.

VM Allocation is a policy of managing the allocation of Virtual Machines to the Physical Node in the Cloud Computing Environment. VM allocation is also termed as VM Provisioning in many literatures. There are many processes involved in VM allocation, namely, Virtual Machine Provisioning, Consolidating, Monitoring, Migrating and Physical Node Selection. VM provisioning is a phase of provisioning VM instances to the users to execute jobs

It is a fair method to allocate several virtual machines to least Physical Nodes as in Bin Packing problem [15] for better resource utilization and minimal power usage. Also, it is necessary to manage the VM count on each Physical Node to avoid degrading VM performance and over utilization of energy [16]. It is suggested in [17] that allocation of many VMs to the same PN considering factors like Service Level Agreements and excessive energy consumption due to over utilization. The heuristic approaches for VM allocations evaluated for Cloud datacenters are First Fit, Next Fit and Least Fit First algorithms. First fit and Next fit were found to be better, in terms of performance. These approaches lack the guarantee of optimal solutions [18], but can be considered for PN selection for VM allocation.

The First Come First Serve (FCFS), Round Robin (RR) and Greedy mechanisms are implemented in Eucalyptus, Nimbus, Open Nebula [19]. In FCFS approach the VM is provisioned on the first available PN that has the capacity to host the VM. It employs First Fit approach for VM provisioning. Eucalyptus uses it to place VM on the first suitable PN. Round Robin allocates VMs on rotation basis in a cyclic manner with the intention of load balancing the PNs. It employs Least Fit approach considering the CPU and RAM capacity. Greedy algorithm identifies the highest utilized PN and assigns a VM to it. It employs Best Fit mechanism. Nimbus uses this mechanism for placement of VM instance but does not employ any live VM migration at runtime which results in better utilization of the resources.

III. METHODOLOGY

The rapid development of the Internet technology and the very drastic nature of heterogeneous clients, applications and tremendous increase in data have imposed a great challenge. To address these challenges Cloud Computing model is the solution. A relook is required for re-provisioning the existing components of the cloud model. In this paper, we propose fault detection and fault prevention methods and policies devised for Light Weight Infrastructure Service Stack

(LWISS) that is resilient to failures. The conceptual model of LWISS stack proposed in [20] is shown in Figure. 2. The stack is devised for low-end commodity hardware, and hence it has been optimized for performance. Debian operating system is chosen with minimal dependencies along with KVM hypervisor, and light weight VMs (MicroVMs) are created for better performance. Fault Tolerance Manager implements the fault detection and prevention policies for VM Creation and job provisioning processes. The methods and policies are derived from Cloud Computing patterns. An experimental test bed was setup with three node low-end commodity hardware. The experimental study on the test bed paved way for the parameters to define the methods and policies. We considered Watchdog -the Cloud computing pattern which regularly monitors the health of the Physical Node and the Virtual Machine at regular intervals and invokes an appropriate trigger when required. The policy decision based on Watchdog pattern in this work is depicted in Figure. 3.

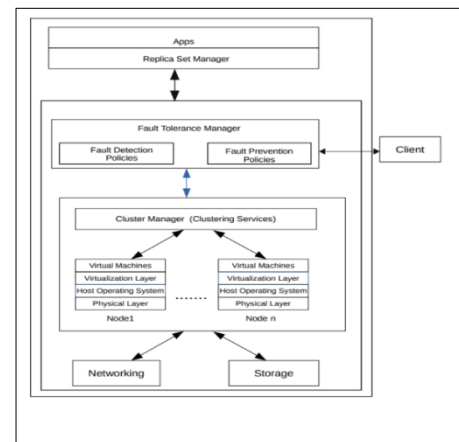


Figure 2: Conceptual Model of the Stack

Two policies are derived for VM Creation Process. We first Detection of VM failures and arrive at a policy to prevent failures by reducing the stress on the Infrastructure as shown in Table: 1 and Table: 2 respectively.

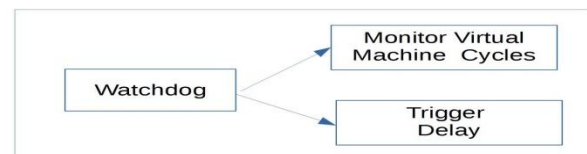


Figure 3: Policy Pattern Mapping

The policies and methods are derived based on the following assumptions (i) It is a private cloud in a trusted environment that is self-hosted. (ii) The deployment adheres to immutable infrastructure (iii) It is a single tenancy model. The work is divided into two stages viz, Virtual Machine Management and Job Submission. The considerations for designing the policies for VM Life Cycle for LWISS are: Detection of VM Failure and Managing the Stress on the Infrastructure.

Table 1. Detection of VM Failure

Vision	Not all VMs of a given service will fail at the same time
Why	To provide reliable application deployment process
How	Monitor the status of the hypervisors within the cluster at regular intervals and maintain symmetry among the nodes
Outcome	Provision availability of services to the app

The policy for detection of VM failure monitors the health of the VMs by regularly checking the uptime and the ping request. If any of the two parameters fail, it triggers a preventive policy and keeps only healthy VMs in the provisioning list.

Table 2: Minimize Stress on the Infrastructure

Vision	Minimal impact on Infrastructure
Why	Use CPU in most favorable manner considering the capacity of the Physical Nodes
How	Introduce delay during VM set creation and monitor the CPU usage and temperature
Outcome	Decision based provisioning of VMs

The second part of the work deals with Job submission. When the user sends a request for deployment, the fault tolerant manager invokes the request and invokes the policy for user request servicing as shown in Table: 3. This policy decides if the user request can be accepted or denied based on the capacity of the resources. If resources are fully used, then job is queued else send notification for resubmission. The job is sent to the queue only if the executing job timing is less than the waiting time of the new job.

The sequence of Job scheduling Work Flow in the LWiSS model is as shown below:

submit Job → *allocate/deny* →

If allocate → *queue/execute* →

If Queue reside in waiting list

If waiting for too long → *resubmit*

If execute → *finish*

If Deny → *finish*

Table: 3 User Request Servicing

Vision	Rule based Service / Denial of user request
Why	Limit overflow of queue and reduce waiting time
How	Queue new jobs based on the resource availability. If the scheduled job completes within in Waiting-time then queue the jobs else deny the request.
Goals/Outcomes	Decision based queuing of jobs

IV. RESULT AND DISCUSSION

The experimental test for creation of Micro VMs was conducted and the impact of VM creation on the infrastructure was observed. Two test cases were considered

(i) creation of 50 Micro VMs and (ii) Creation of 100 Micro VMs. Each test case was tested under three conditions without delay, with 3 min delay and with 5 mins delay respectively. Test was conducted for five runs each. The observation was that a delay of 3 mins was an ideal condition for both the test case and the same is depicted in Figure: 4 and Figure:5 respectively.

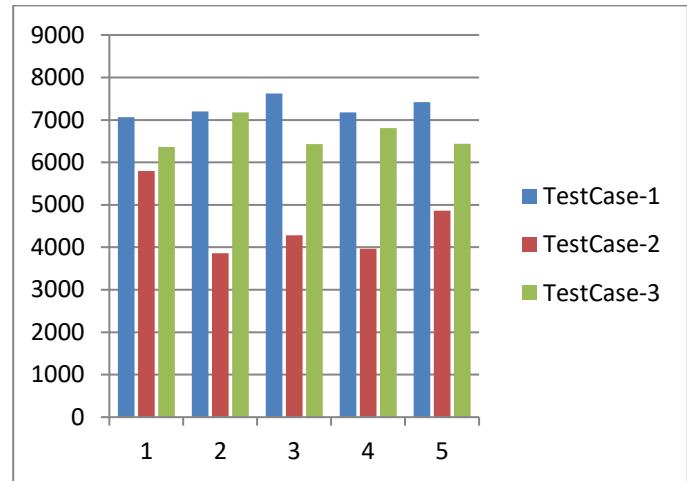


Figure: 4 Time to create 50 VMs

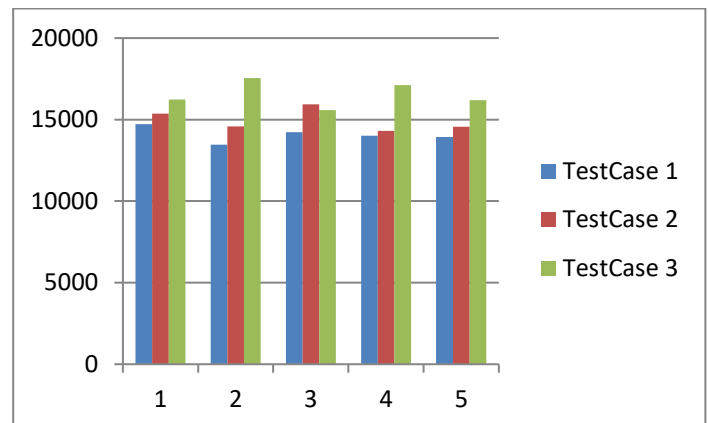


Figure: 5 Time to create 100 VMs

Based on these experimental study and infrastructure optimization, we have arrived at two decision tables for Node Capacity and Job Scheduling as shown in Table: 4 and Table: 5 respectively. The submitted task will be deployed based on the decision made by the policies during the deployment stages. Table: 6 shows the queue stages in the system. The fault tolerant manager and the decision policy are depicted. Policy decision is for VM creation based on decision table 1 and allocation is made considering the node capacity available at that given point of time. Jobs requests are accepted or denied based on the decision tree for job scheduling. These decision tables provide resilient computing environment to handle failures in a proactive manner.

Table 4: Decision Table for Node

Node_Id	Cores (Nos.)	RAM (GB)	Disk (GB)	VM Density (Nos.)
N1	3	8	100	30
N2	5	8	100	50
N3	7	16	100	100

Table 5: Decision Table for Job schedule

Job_Id	VM Cycle type (mins)	App Life Cycle (mins)	Jobs (Nos.)
J1	5	30	6
J2	10	30	3
J3	15	30	2

Table 6 : Decision Table for Waiting Jobs

Node_Id	Cores	Queue size	Queue State	Max. Jobs
N1	3	1	0/1	10
N2	5	1	0/1	10
N3	7	2	0/1	20

V. CONCLUSION

The current day user demands and the nature of applications make cloud computing an indispensable computing model of the present day needs. Providing Fault tolerance computing is one of the key requirements. This paper proposes fault tolerant policies for detecting VM failure and policy to reduce the stress on the Physical Node. Decision tables are drawn from the experimental tests conducted for determining the capacity of the infrastructure. The policies are based on the decision tables. Along with scheduling algorithms for VM scheduling, task scheduling and VM allocation policies are also discussed. The LWISS stack aims at providing failure resilient computing environment. Low-end commodity hardware has been used for the work keeping in mind reusability of existing resources. The policies together make the environment resilient to failures and thus act proactively to any faults rather than wait for the failures to occur.

REFERENCES

- 1 Mesbahi, M.R., Rahmani, A.M. & Hosseinzadeh, M. "Reliability and high availability in cloud computing environments: a reference roadmap". *Humanities Centric Computing Information Science*. 8, 20 (2018) doi:10.1186/s13673-018-0143-8.
- 2 Ardagna D (2015)," Cloud and multi-cloud computing: current challenges and future applications", 7th International workshop on principles of engineering service-oriented and cloud systems (PESOS) 2015. IEEE/ACM, Piscataway, pp 1–2.
- 3 Mell P, Grance T (2011),"The NIST definition of cloud computing". *Commun ACM* 53(6):50.
- 4 Buyya R et al (2009)," Cloud Computing and Emerging IT Platforms: vision, hype and reality for delivering computing as the 5th utility. *Future Gener Comput Syst* 25(6):599–616.
- 5 Puthal D et al (2015)," Cloud computing features, issues, and challenges: a big picture". In: International conference on

computational intelligence and networks (CINE). IEEE, Piscataway, pp 116–123.

- 6 Motavaselalhagh F, Esfahani FS, Arabnia (2015), "Knowledge-based adaptable scheduler for SaaS providers in cloud computing", *Hum Cent Comput Inf Sci* 5(1):16
- 7 Alain Tchana, Laurent Broto and Daniel Hagimont, "Approaches to Cloud Computing Fault Tolerance," in International Conference on Computer, Information and Telecommunication Systems (CITS), France, pp. 1-6, 2012.
- 8 M. R. Lyu , "Software Fault Tolerance", New York:Wiley, 1995.
- 9 C. Binnig, D. Kossmann, K. T., and S. Loesing, "the Weather tomorrow?: towards a benchmark for the cloud," in Second International Workshop on Testing Database Systems, 2009.
- 10 Q. Cao, W. Gong and Z. Wei, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing," In Proceedings of Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp. 1-3
- 11 Dr. Sudha Sadhasivam, R. Jayarani, Dr. N. Nagaveni, R. Vasanth Ram "Design and Implementation of an efficient Two- level Scheduler for Cloud Computing Environment" In Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing, 2009
- 12 G. Guo-Ning and H. Ting-Lei, "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment," In Proceedings of International Conference on Intelligent Computing and Integrated Systems, 2010, pp. 60-63
- 13 Anders Mikkelsen, Tor-Morten Grønli, Rick Kazman, "Immutable Infrastructure Calls for Immutable Architecture: Deploying a Changeless Architecture in the Cloud", 52nd Hawaii International Conference on System Sciences 2019, URI: <https://hdl.handle.net/10125/60142>, ISBN: 978-0-9981331-2-6.
- 14 C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE Softw.*, vol. 33, no. 3, pp. 94–100, 2016.
- 15 M. Callanan and A. Spillane, "Devops: Making it easy to do the right thing," *IEEE Softw.*, vol. 33, no. 3, pp. 53–59, 2016.
- 16 H. Kang, M. Le, and S. Tao, "Container and microservice driven design for cloud infrastructure devops," in 2016 IEEE International Conference on Cloud Engineering (IC2E), 2016.
- 17 R. Buyya, J. Borberg and A. Goscinski, "Cloud Computing Principles and Paradigms", Wiley Publications, 2011.
- 18 W. Shi and B. Hong, "Towards profitable virtual machine placement in data center", 2011, 4th IEEE International conference Util Cloud Computing, Dec, 2011.
- 19 K. Mills, J. Filiben and Dabrowski, "Comparing VM-Placement algorithms for on-demand clouds", 2011, IEEE 3rd Int. Cnf on Cloud Comp. Tech. Sc. Pp.91-98, 2011.
- 20 Sudha M, Usha. J, "A Novel Fault Tolerant Approach using Patterns for Private Cloud Environment", *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 17, No. 7, July 2019,

AUTHORS PROFILE



Sudha M. is presently pursuing research at RVCE-MCA research Centre, Visvesvaraya Technological University, Bengaluru in the area of Cloud Computing. She has 24 yrs of work experience which includes, five years of teaching and seven years of research experience. Her areas of interest include cloud computing, distributed computing, networking, databases. She has published ten

papers in reputed international conferences and Journals.



Usha J is working as Professor, Dept of Master of Computer Applications (MCA) Department, RV. College of Engineering, Bengaluru. She obtained her B.Sc and MCA degree from Bangalore University. She has over Twenty two 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.