ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Experimental Investigation Decentralized IaaS Cloud Architecture Open Stack with CDT

S. Gobinath, S. Saravanan

PG Scholar, CSE Dept, M.Kumarasamy College of Engineering, Karur, India[1]

Assistant Professor, Department of CSE, M.Kumarasamy College of Engineering- Karur, India[2]

ABSTRACT: Investigating decentralized advanced the virtualization technology in data centres lead to the development of features as High availability, Disaster recovery and Fault tolerance. By using the Amazon Elastic compute cloud (EC2) and tools, IaaS- Infrastructure as a service computing clouds that span across distributed areas require more advanced mechanism. The reSsearch will be done in decentralized IaaS cloud architecture using the concept of open stack pioneered latency. Additional features also providing to safeguard the Infrastructure and our data by introducing Continuous Disaster Tolerance (CDT).

KEYWORDS:Decentralized, Disaster Tolerance

I. INTRODUCTION

Disaster tolerance (DT) refers to the capability of a computing and data environment to survive a disaster (such as loss of communication components, loss of power and a human-made or natural catastrophe) and to continue to function, or to return to its functionality in a relatively short period of time [8]. DT capability requires a distributed system with redundant elements that are physically separated on distances ranging from few buildings to continents. The virtualization technology opened a new way to control and manipulate the operating systems that run inside the virtual machines (VM). Available techniques allow hypervisors to live migrate a VM between hosts preserving clients' connections in a transparent manner [4]. Commercial grade open source hypervisors, like KVM and XEN, are available as support for advanced features. Cloud computing becomes gradually the computing infrastructure of the next decades. One type of such clouds, the Infrastructure as a Service (IaaS), delivers a virtualized computer infrastructure employing hypervisors' capabilities. One of the most successful IaaS to date is the Amazon Elastic Compute Cloud (EC2) which implements advanced computing and storage capabilities [13]. It runs multiple clusters across the globe, grouped in what they call availability zones. Open source IaaS clouds have reached a good maturity level. Such an example is Eucalyptus project which made its way into Ubuntu Server distribution [6].Open source virtualization toolkits that manage hypervisors and controls VMs lifetime, like the Livbirt project, stays at the foundation of IaaS clouds. This paper presents an end-to-end Disaster Tolerance (DT) enablement solution for IaaS clouds. The paper is organized in seven sections. The second section presents the state of the art. The third section introduces the Qemu-kvm DT system detailing its building blocks: the Seven-stage DT algorithm, the Romulus hypervisor implementation, the DT API and scenarios. The forth section highlights the virtualization toolkit DT API and framework. The fifth section presents the DT enabled EC2 API, the Eucalyptus hierarchical IaaS cloud architecture, allocation algorithms and failover approaches. The sixth section summarises the reference implementation aspects of the DT enablement solution. The last section addresses the conclusions and the areas of further research.

II. EXISTING SYSTEM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

We gently introduced Romulus in [3], a DT solution that addresses the issues identified in Remus reference implementation. The goal of this section is to depict the core algorithm for building reliable DT systems and to detail the DT API and DT scenarios built on top of Romulus implementation.

1 Seven-stage DT algorithm

The seven-stage DT algorithm which stays at Romulus's core involves two hosts, the primary on which a VM is running and the backup with an idle VM. The hosts may be placed in remote locations, and must be served by a high speed connection.

Stage I - Disk replication and Network protection (Fig. 1)

- Primary host buffers network egress traffic
- Primary host applies local disk writes
- Primary host replicates disk writes
- Backup host buffers replicated disk writes

Stage II – Virtual Machine checkpoint (Fig. 2)
- Replication synchronization handling is a prerequisite
- Primary host suspends VM
- Primary host copies VM delta state to a buffer
- Primary host requires checkpoint synchronization

Stage III - Checkpoint synchronization (Fig. 3)
- Primary host creates a new buffer network traffic
- Backup host creates a new disk buffer
- Primary host resumes VM

Stages IV/V – Second Disk replication and Network protection, VM replication (Fig. 4)
- Primary host routes egress traffic to the new buffer
- Backup host routes replicated disk writes to the new Buffer
- Primary host replicates VM buffer on a dedicated Thread
- Backup host buffers the received VM state
- Backup host requires replication synch when VM replication completes

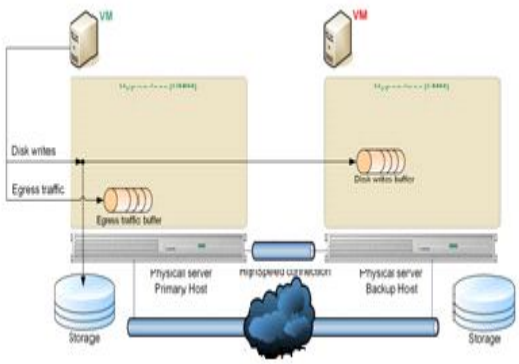

Fig 1:The Seven-stage DT algorithm Stage I

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

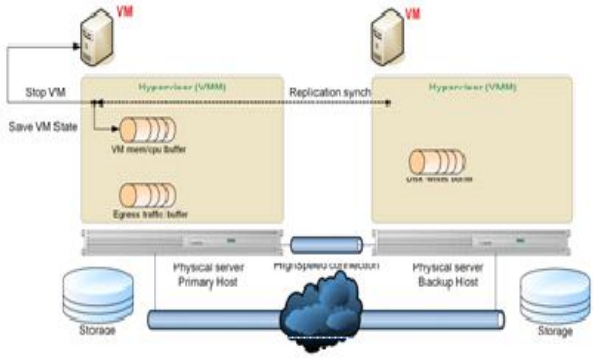

Fig 2: The Seven-stage DT algorithm Stage II

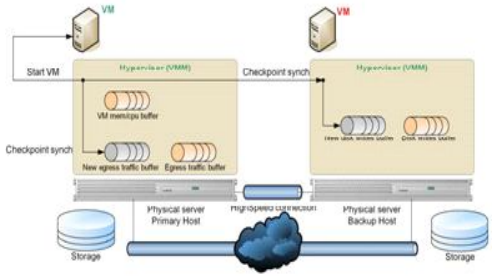

Fig 3: The Seven-stage DT algorithm Stage III

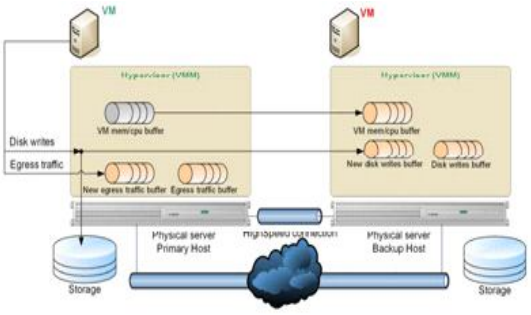

Fig 4: Seven-stage DT algorithm Stage IV/V

Stage VI - Replication synchronization (Fig. 5)
• Primary host releases network buffer.
• Primary host makes new network buffers current.
• Backup host flushes VM and disk buffers.
• Backup host makes new disk buffer current.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Stage VII - Failure detection and failover (Fig. 6)
• Backup host monitors primary host heartbeat.
• Backup host detects primary host failure, executes remaining.

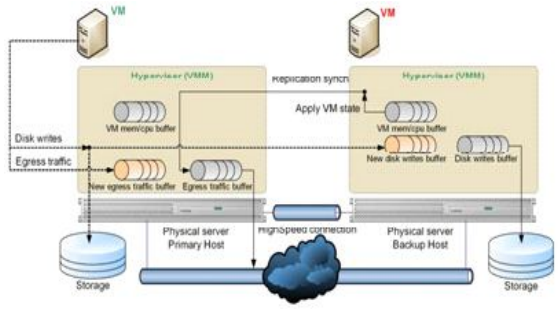

Fig 5: The Seven-stage DT algorithm Stage VI

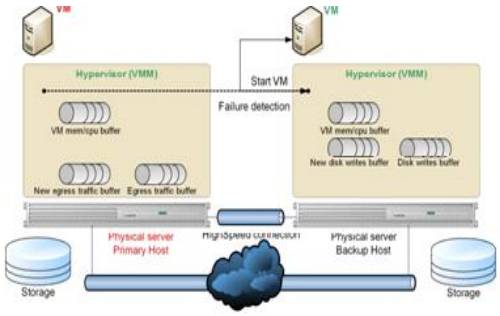

Fig 6:The Seven-stage DT algorithm Stage VII

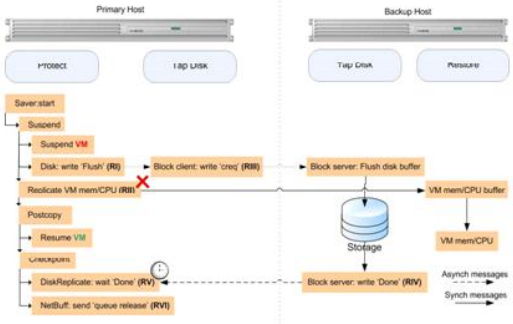

Fig 7:Remus Execution Flow

2 Romulus Implementation

Romulus was implemented as a DT feature in open source KVM hypervisor and its companion Qemu virtualizer [9], [7].

The following Qemu building blocks where tailored to accommodate the DT functionality

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

- Live replication algorithm extended with continuous high frequently replication.
- Block driver model accommodated a proxy
  Driver to handle disk replication.
- Network egress traffic benefited from a
  Simplified proxy driver model.

3 Comparative results

We composed our seven-stage DT algorithm and Romulus implementation performance with the only solution that resembles its capabilities [5].

Remus runs two main processing blocks on either side of the network, each of them working interconnected.(fig.7)

III. PROPOSED SYSTEM

1 DT Virtualization

For the reference implementation we selected libvert [11], a renowned open-source virtualization toolkit.Libvirt supports the Qemu-Kvm virtualizer and offers remote management by using authentication, encryption and certificates.

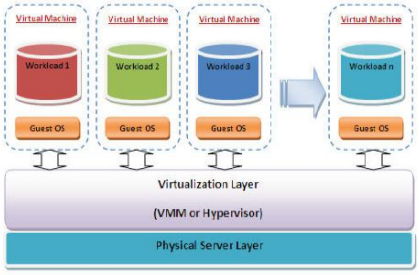

Fig 7:Virtual Environment with multiple VM's

2 DT in the Decentralized IaaS

The concept of decentralized will overcomes the usages of both primary and backup sites in the IaaS cloud architecture. By using the concept of decentralized there will no centre it will be present in world wide area.(fig. 8)

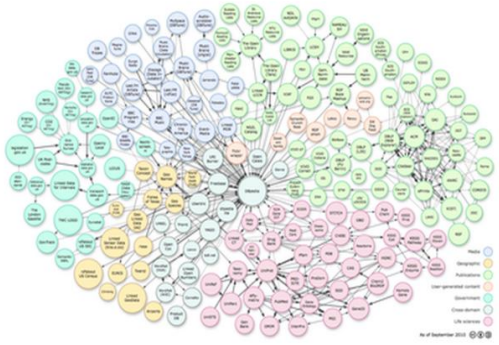

Fig 8: Decentralized IaaS DT Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

3 DT Pioneered lately

Disaster tolerance mainly focuses on the pioneers status because of our previous latency techniques and also it's queuing. For processing in the queue we proposed an open stack queuing theory method and it mostly very helpful for the pioneers in demand to access the cloud IaaS Architecture.

4 DT in the Cloud

In cloud the major usages will be present without any interruptions.(fig. 9).

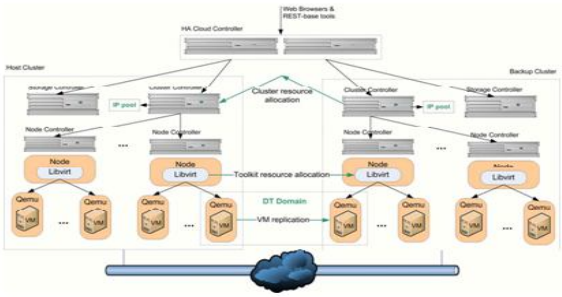

Fig 9: DT Enabled Cloud Architecture

DT may be occurs due to poor bandwidth connections also tolerated.

5 Cloud Tools used

-open source Kvm

-Amazon EC2 SOAP, euca2tools and boto

-Eucalyptus cloud, cluster and Node controllers

IV. CONCLUSIONS

In this paper we present an Disaster Tolerance (DT) in Decentralized IaaS cloud architecture. Also an experimental verification solution will be provided for our cloud Infra structure.[14]

In conclusion we provided the pioneered latency techniques for worldwide area. In order to detect the priority we are also considering the queuing method. The open stack queuing will be integrated with the pioneered latency modelling.

REFERENCES

[1] Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris, T.; Ho, A.; Neugebauery, R.; Pratt, I.; Warfield, A.; "Xen and the Art of Virtualization" SOSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003.

[2] Brendan, C.; http://dsg.cs.ubc.ca/remus, UBC, 2009.

[3] Caraman M, C; Moraru, S, A; Dan, S, Kristaly, D, M; "Romulus, Disaster Tolerant System based on Kernel Virtual Machines", The 20th International DAAAM Symposium "Intelligent Manufacturing & Automation: Theory, Practice & Education", 25-28th November 2009, Vienna, Austria.

[4] Clark, C.; Fraser, K.; Hand, S.; Pratt, I. & Warfield, A. "Live migration of virtual machines", Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation, 2005.

[5] Cully, B.; Lefebvre, G.; Meyer, D.; Fraser, K.; Hutchinson, N. & Warfield, A.; "Remus: High Availability via Asynchronous Virtual Machine Replication", Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, 2008

[6] Daniel, N; Rich, W; Chris, G; Graziano, O; Sunil, S; Lamia, Y; Dmitrii, Z; "The Eucalyptus Open-source Cloud-computing System", in Proceedings of 9th IEEE International Symposium on Cluster Computing and the Grid, Shanghai, China 2009

[7] Fabrice, B; "QEMU, a Fast and Portable Dynamic Translator", Proceedings of the annual conference on USENIX, Annual Technical Conference, Berkeley, CA, USA 2005

[8] HP, "Delivering high availability and disaster tolerance in a multi-operating-system", HP, 2006

[9] Kivity A.; Kamay Y.; Laor D.; Lublin U.; Liguori A., "kvm: the Linux Virtual Machine Monitor", Proceedings of the Linux Symposium,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Ontario, Canada 2007
[10] http://www.vmware.com/products/fault-tolerance, VMware Fault Tolerance, VMware, 2011
[11] http://libvirt.org, Red hat, 2012
[12] http://code.google.com/p/boto/, Google Code, 2011
[13] http://aws.amazon.com/documentation/ec2/, Amazon, 2012
[14] http://openstack.org, Open Stack, 2012.