A Survey on Enhancement Cloud Portability with Basic Utility

Mr.S. Ananth[1], Asst.Prof.
Department of Information Technology
Mahendra Engineering College
Namakkal, India.

Mrs.K.Radha[2],Asst.Prof,
Department of Information Technology
Mahendra Engineering College
Namakkal, India.

Miss.S. Sowndarya[3]
B.Tech- Final year
Department of Information Technology
Mahendra Engineering College
Namakkal, India.

Miss.M. Shyamala[4]
B.Tech- Final year
Department of Information Technology
Mahendra Engineering College
Namakkal, India.

Today's IT services are moving to cloud computing environment in order to process client request and provide services effectively. In this case, reliability is a major factor to reduce the load in storage and network resource. In traditional method, it uses fault tolerance, replication of VM and storing checkpoint image in a neighboring server. Replication increases cost for a large system and checkpoint image inaccessibility occurs if any node crashes. It can be rectified by distributed storage of checkpoint across every server. This persistent storage can be made using a heuristic algorithm – an optimization problem. These experiments are verified using a cloud simulator. It does not provide effective results on reliability. Thus, in this proposed system a real cloud environment is chosen – OpenStack to verify experiments and to estimate the performance.

Index Terms **– Reliability, Checkpoint images, Openstack, Data center, Node**

I.INTRODUCTION

A.Cloud Computing

In today's IT environment cloud computing is becoming a basic utility. It provides various services to clients at any time and any ware. [1] Some of the major services are

- PaaS (Platform As a Service)
- IaaS (Infrastructure As a Service)
- SaaS (Software As a Service)

Results of survey have introduced two new types of services. They are

- Reliability as a Servie (RaaS)
- Fault Tolerance as a Service (FaaS).

There are mainly three types of cloud computing available. They are

- Public
- Private
- Hybrid cloud.

Each cloud provides all services based on demand and through payment. An open and scalable Operating System is available to build public and private cloud, which is called as OpenStack

B.OpenStack

It is used for both large and small organizations and reduces the risk. It is also called as Cloud Operating System. [2] Some of the similar OS are CloudStack, OpenNebula and etc. It has various components. Some of them are

a) Nova

Nova is a first engine in OpenStack. It mainly used to deploy and manage large number of virtual instances. It also manages computing tasks.

b) Swift

Swift is a storage device to store objects and files. It has the capacity of scaling easily. It manages backup if any fault occurs.

c) Cinder

It is mainly made for block storage. It retrieves files easily and in a faster manner from any location in a disk.

d) Neutron

Networking is major in these aspects. It can be getting through neutron. It is used to build a communication with each other components.

e) Horizon

It gives Graphical User Interface to user for easier access. It allows admin to look the activities of each. It can be accessed by API.

f) Keystone

It lists the user in OpenStack and provides an identity. Each services are mapped with particular resource based on their need.

g) Glance

It takes the snapshot of the state of virtual machine and stores for future use, in case of any fault or failure of a virtual machine.

h) Ceilometer

Each service can be get through payment basis based on user request. This billing is done by ceilometers.

i) Heat

It stores the files which is required for cloud application and resources for the one. It mainly used to manage the infrastructure.

It includes various versions, they are

TABLE.1 VERSIONS OF OPENSTACK

VERSION NAME	RELEASE DATE	COMPONENTS INCLUDED
Austin	21st October 2010	Nova, Swift
Bexar	3rd February 2011	Nova, Swift, Glance
Cactus	15thApril2011	Nova, Swift, Glance
Diablo	22nd September 2011	Nova, Swift, Glance
Essex	5th April 2012	Nova, Swift, Glance, Horizon, Keystone
Folsom	27th September 2012	Nova, Swift, Glance, Horizon, Keystone, Quantum, Cinder
Grizzly	4th April2013	Nova, Swift, Glance, Horizon, Keystone, Quantum, Cinder.
Havana	17th October 2013	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat.
Icehouse	17th April 2014	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove.
Juno	11th October 2014	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder,Ceilometer, Heat, Trove, Sahara.
Kilo	4th April 2015	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove, Sahara, Ironic.
Liberty	16th October 2015	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove, Sahara, Ironic, Zahar, Manila, Designate, Barbican.
Mitaka	7th April 2016 (Under development)	Nova, Swift, Glance, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove, Sahara, Ironic, Zahar, Manila, Designate, Barbican.

Still two more components are under development. They are

- N*
- O*

C.Reliability

Reliability is nothing but the ability to perform without failure. In large scale cloud services, clients request are serviced with high reliability. It is a major factor in case of node crash. This is due to

- Server down
- Heavy load
- Hardware failure
- Natural Disaster

Whenever a Virtual Machine fails its services should not be affected. This paper lists various techniques to improve reliability.

II.TYPES OF RELIABILITY ENHANCEMENT

There are fewer models are employed to improve the reliability. They are

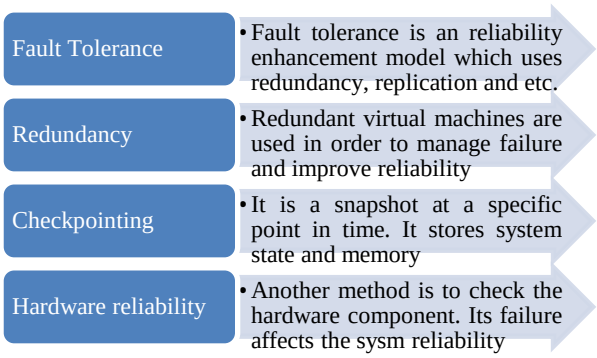

Fig. 1 Reliability enhancement types [1]

III.LITERATURE REVIEW

A. Reliability Enhancement with Optimal Resource Usage

In order to avoid data loss and unavailability checkpoints are stored in a different manner. [3] It uses two types of checkpoints. First is Service checkpoint, which is used to store the similar data for once. Second is Delta Checkpoint, which is taken periodically whenever a small modification is made on the data. These checkpoints are stored globally for faster retrieving process. An efficient optimization algorithm called heuristic algorithm is employed to manage the storage problem.

B. A Byzantine Fault Tolerance Framework

This paper ensures high reliability of services which is conducted in a real environment with 257 cloud providers. [15]In this framework, only if there are more than three faulty machines then it can be replaced with a new one. This replacement takes five steps. First it selects a node, replicates in multiple set of VM, client requests are executed in all sets and also in primary node, updates the modifications in all node and respond to client or cloud module. If any failure occurs in update it can be rectified by changing the modifications. It replaces a machine only if the entire three replicas get crash. This makes the system slowdown to respond the client request. This may lead to data inaccessibility if primary copies get crash and it is not recovered. Thus, it cannot be applicable for larger clusters.

C. Redundant Virtual Machine placement

Main aim is to reduce the number of host servers. It is achieved using multiple redundancies in a host server. [6]This reduces the failure rate and fault. In case of fault tolerance it needs three number of virtual machine. Even though the capacity is increased this k-redundancy is not improved. Results are gathered for both k-redundant and multiple k-redundant with minimum number of hosting server. In these methods, difference between two are very small increasing the capacity. In multiple k-redundant same applications is placed in different virtual machine in a same hosting server.

D. Component ranking for Fault Tolerance

In this paper, it uses mainly two algorithms, first one is to define the structure and design of the component and the other is to identify the component which is more faulty and failed. [16]This experiment uses five concepts, NoFT, RandomFT, FTCloud1, FTCloud2, AllFT. As the name implied in NoFT, it does not have any fault tolerance. RandomFT has only for k number of components. In FTCloud1 it ranks the information of structure and FTCloud is for hybrid component and it require basic idea to implement. Faults can be identified automatically. Experiments proved that the faults with lesser defects are removed to improve the reliability of the service. Thus the ranking is used to choose the component based on defect rate.

E. VM level Fault Tolerance

In order to avoid failures in node, it takes checkpoint images which is nothing but snapshot of the system state. [9] This checkpoint is taken for entire virtual cluster which includes routing, backup storage, node usage and etc. It helps to recover the machine state easily without any loss of

data and in a faster manner. Whenever node crash occurs it can be rectified easily with help of saved machine state by recovering. Xen Architecture is used to conduct the experiments. Though it provides better and good results for reliability, cost for simpler model is high. Thus it cannot be employed for smaller experiments.

F. Checkpoint based Fault Tolerance

A new file system called Another Union File System is used to take checkpoint of read-only data. [7]It is taken only once in another case, read-write data are checkpointed for even small modification. AUFS takes lesser time to take the checkpoint. These checkpoints or system states can move to Hadoop Distributed File System. A use of HDFS is to automatically replicate the snapshots in various instances or VM. It reduces the resource usage to replicate. It also provides suspend mode when node failure occurs and recovers from a replica which is stored in HDFS. The main problem in this method is when retrieving the state from HDFS it takes comparatively large amount of time to continue the suspended process.

G. Fault Tolerance for a System level Perspective

In this paper, it introduces a new service called Fault Tolerance as a Service. In which, it allows user to define the fault tolerance level through service layer. [12] Every client can get their level based on the requirements of the application. It does not require any knowledge to define the level. It employs various managers like, Replication, Fault Masking, Fault Detection and Recovery Manager. Each one is assigned with separate task to make faster access and to improve the performance by increasing reliability. It uses web services like WS-RM and WS-BPEL. It results in average runtime overhead.

H. Fault Tolerant Layer 2 Data Centre

Due to increase in number of clients, resource usage also increased. It can be managed through scalable resource. [11]Scalable resource is achieved by layer 2 routing in Data Centre and through multirouting tree topology. This solves the problem of less scalability through hybrid based arbitrary topology. In case of failure in node crash services are given with help of Virtual Machine migration. But it cannot be applied to more number of accesses as it does not visible due to scale. Thus it degrades the system performance.

I. Reliability as an elastic Service

This can get through checkpointing the VM and its state. In addition to checkpoint it also stores the history of action which helps in failure rate estimation denoted by λ. [10]It uses peer to peer checkpointing mechanism. Under-utilized resources are utilized through request. This mechanism is employed for all resources and accessed with help of distributed algorithm by using dual decomposition. It increases performance and reliability comparing with centralizedcheckpointing. Problem in this method is that it cannot be employed for larger system.

J. Hardware Reliability

As todays IT organizations are moving to cloud computing, scalable resource provisioning methods are employed. [8] These resources satisfy the user needs through various models. In this case, reliability is a major impact to get access without any failure. Scalable resource may slow down or damage the hardware components which in turn make less reliable service. This can be rectified by characterizing the hardware failure before it leads to loss of access. Characterization is based on number of hard disk, its life, manufacturer, reason for failure, location and etc. By finding this information, faults can be rectified. Though it manages everything, changing the failed hardware with newer one takes large amount of time and cost. Thus it is not reliable for large data centres.

K. Reliable Resource Provisioning policy

It mainly deals with increasing the availability of system. [13]Reliability is enhanced by failure rate awareness and rules. Correction of failure is made using temporal and spatial factors. It ensures the quality of service, performance, reliability and efficient resource access.

L. Reliability based Optimization for cloud migration

In this framework, mainly it uses two ranking algorithm. [14]First one is to move all components to cloud and the second is to move only the part of the component (modifications). It improves

reliability by fault tolerance strategy. Method used here is that only the lesser defects are replaced than the greater defects. This is made using optimization framework. This uses a software based fault tolerance strategy. Results show that the smaller threshold is to be refactored again. This makes the system to degrade and affects the performance in case of more failures.

It mainly deals with scheduling of resources for clients in a reliable manner. Virtual machine provides IaaS to the clients. [4]VMs resource is organized in an efficient way for better reliability. Calculating the reliability includes data centre and virtual machine reliability. By comparing the results, determination is made to concentrate in which part of the system.

M. **Reliability as a Service**

TABLE.2 COMPARISON TABLE

PAPER TITLE	METHODOLOGY USED	MERITS	DEMERITS
On Cloud Service Reliability Enhancement with Optimal Resource Usage	It uses 1. Service checkpoint 2. Delta checkpoint	It stores checkpoint globally, which avoids loss of data.	It uses a simulation tool to conduct experiments. Results are not accurate.
BFTCloud: A Byzantine Fault Tolerance framework for voluntary – resource Cloud Computing.	Experiments are made in a real environment with 257 cloud providers. It considers that the particular VM is failed only if all the three replicas are failed.	It ensures reliability in a real model with efficient resource usage.	It causes the system to degrade and loss of data may occur.
Redundant Virtual Machine Placement for Fault Tolerant Consolidated server cluster.	It uses Multiple redundancy to reduce the host server for replication of VM	Redundant copies helps in avoiding data inaccessibility.	Its performance is better but not accurate
Component Ranking for Fault Tolerant Cloud Applications.	Two algorithms are employed. Used for 1. Component structure 2. Identifying failure component	Identification makes easier to replace the failed component with the newer one.	Defining structure is difficult if does not have knowledge about it
Checkpoint based Fault Tolerant Infrastructure for Virtualized Service Providers.	Checkpoints are used to recover the machine state, which is stored in central storage server. 1. System Checkpoint 2. Delta Checkpoint	It stores the checkpoints in a host server. It reduces the disk space.	When the host server fails the reliability cannot be achieved.
Fault Tolerance Management in Cloud Computing: A System Level Perspective.	A separate service layer is used to allow user to define the fault tolerance level. It does not require knowledge	It lists the possibilities of reliability problem and how it can be avoided.	If any link disconnects then it does not recover the state
PortLand: A Scalable Fault Tolerant Layer 2 Data Centre Network Fabric.	It uses multirouting tree topology to increase the availability of resource through which reliability can be enhanced	Core switches are used for moving the traffic congestion	Migration of larger is not visible due to its scale

V. PROPOSED WORK

In this proposed work, Check points are stored in two places. They are global storage and in number of Virtual Machines. It overcomes the demerits called Data inaccessibility due to node crash. It enhances reliability by increasing the response time though when the virtual machine failure occurs and increases the speed of the accessibility by reducing the time. This can be done with help of characterizing the checkpoints as Service checkpoint and Delta checkpoint. Service checkpoint is taken only once and Delta checkpoint is taken for every modification in the state of the system. This reduces the overflow and traffic in a network thereby increasing the time of recovery. It helps in restoring the state of a failed machine

Tool used to implement the work is OpenStack. It provides Infrastructure as a Service.

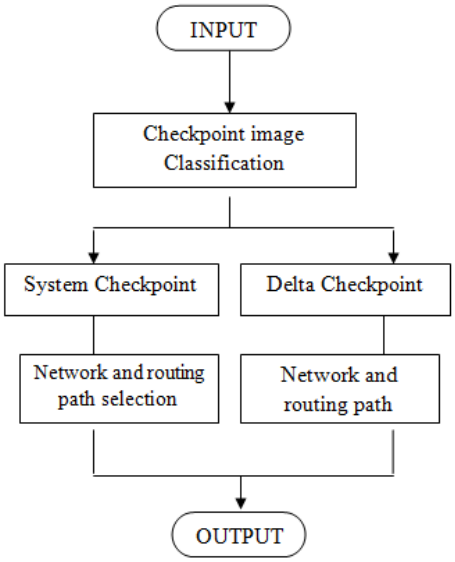

Fig. 2 Architecture Diagram.

VI.IMPLEMENTATION

This experiment is made in an OpenStack environment. A model called Devstack is used to set up OpenStack framework. It uses Python as a programming language. Steps included to experiment are:

- Create an OpenStack Environment
- Creating and Launching instances
- Creating Checkpoint images
- Storing Checkpoint images
- VM migration

A. Creating an OpenStack Environment

It uses Devstack to create OpenStack environment. It has many packages. It is a script used to install a framework in an easier manner. It has various inbuilt components linked together are Nova, Keystone, Cinder, Neutron and etc. It is a basic platform setup to build services and develop them.

B. Creating and Launching instances.

Instances act as a single VM which renders services to client request. It is made using the details like,

- Name
- Virtual CPU
- RAM
- Root Disk
- Ephemeral Disk
- Swap Disk

C. Creating Checkpoint Images

Checkpoint images are used to take a backup copy of a machine's current state. It is made to take in a scheduled period of time.

- **Service Checkpoint**

It is made to take only once for storing the read only data of machine. It is stored in a global machine and in all other virtual machines for first time.

- **Delta Checkpoint**

The threshold value of Delta checkpoint is maintained in a slight variation of the latest checkpoint taken. It takes continuous copy with change in the state of the machine.

D. Storing checkpoint images

Once images are created, it is stored in two different locations. They are

- Global – Host or server
- Local – All VM

E. VM Migration

Once the system detects the failure in any Virtual machine, it can be replaced with the latest copy of checkpoint. The checkpoint for replacement is chosen by comparing with all copy and uses the state with last accessed. It helps in recovering the data. This VM migration after failure of particular node is made with help of Neutron (provides Networking).

VI.CONCLUSION

The main aim is to improving the reliability in various aspects with increasing in user request and scalable resource. Routing algorithms are employed to reduce the traffic rate.

In this paper, various techniques for enhancing reliability of the cloud environment are discussed. This survey helps to understand the provisioning model used in a real environment and their related results.

This methodcan also be used in a map reduce model which enhances reliability by faster retrieval of data that the client is requesting.

VII. FUTURE WORK

In future, this method can be designed for one of the Cloud services called as MapReduce framework. This uses SAHARA. It act as an interface for OpenStack.

REFERENCES

[1] "Types of cloud services," *http://www.appcore.com/3-types-cloud-service-models/*

[2] "Various components of OpenStack with versions,"*http://docs.openstack.org/arch-design/content/openstack-components-arch-storage.html*

[3] A. Zhou, S. Wang, Z. Zheng, C. Hsu, M. R. Lyu, and F. Yang, "On Cloud Service Reliability Enhancement with Optimal Resource Usage," *IEEE Transactions on Cloud Computing, Manuscript ID.*

[4] A. Chowdhury, and P. Tripathi, " Enhancing Cloud Computing Reliability using Efficient Scheduling by Providing Reliability as a Service*," Int'l Conference on Parallel, Distributed and Grid Computing.*

[5] "An Introduction to OpenStack," *https://wiki.openstack.org/wiki/Release_Naming*

[6] F. Machida, M.Kawato, and Y. Maeno, "Redundant Virtual Machine Placement for Fault Tolerant consolidated server Cluster*," Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS'10), pp. 32-39, Apr. 2010.*

[7] I. Goiri, F.Julia, J. Guitart, and J. Torres, "Checkpoint based Fault Tolerant Infrastructure for Virtualized service Providers," *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS'10), pp. 455-462, 2010.*

[8] K. V. Vishwanath, and N. Nagappan, "Characterizing Cloud Computing hardware Reliability," *Proc. the 1st ACM Symposium Cloud Computing (SOCC'10), pp. 193-204, Jun. 2010.*

[9] M. Zhang, H. Jin, X. Shi, and S. Wu. 2010, "Virtcft: A transparent vm-level Fault-tolerant system for virtual Clusters," *Proc. IEEE 16th Int;l Conf. Parallel and Distributed Systems (ICPADS'10), pp. 147-154, 2010.*

[10] N. Limrungsi, J. Zhao, Y.xiang, T. Lan, H.H. Huang, and S. Subramaniam, "Providing Reliability as an elastic service in Cloud Computing," *Proc. IEEE Int'l Conf. Communications (ICC'12), pp. 2912-2917,2012.*

[11] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A Scalable Fault Tolerant Layer 2 Data Centre Network Fabric," *Prpc. ACM Int'l Conf. Data Commuicatiion [SIGCOMM' 09] pp. 39-50, Aug. 2009.*

[12] R. Jhawar, V. Piuri, and M. Santambrogio, "Fault Tolerance Management in Cloud Computing: A System Level Perspective," *IEEE Systems Journal, Vol. 7, no. 2, pp. 288-297, 2012.*

[13] T. Guanhua, M. Dan, and Z. Jianfeng, "Reliable Resource Provision Policy for cloud Computing," *Chinese Journal of Cpmputers, vol. 10, no. 33, pp. 1859-1872, 2010.*

[14] W. Qiu, Z. Zheng, X. Wang, X. Yang, and M.R. Lyu, "Reliability-Based Design Optimization for Cloud Migration," *IEEE Transactions on Service Computing, vol. 7, no. 2, Apr. Jun.2014.*

[15] Y. Zhang, Z. Zheng, and M. R. Lyu, "BFTCloud: A Byzantine Fault Tolerance framework for voluntary resource Cloud Computing*," Proc. IEEE Int'l Conf. Cloud Computing (Cloud' 11), pp. 444-451, Jul. 2011*

[16] Z. Zheng, T. Zhou, M. Lyu, and I. King, "Component Ranking for Fault Tolerant Cloud Applications," *IEEE Transactions on Service Computing, Vol. 5, no. 4, pp. 540-550. 2012*