Efficient Live Virtual Machine Migration In Open Cloud Platform

Prepared By

Pradip Patel

12MCEC42

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2014

Efficient Live Virtual Machine Migration In Open Cloud Platform

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

Prepared By

Pradip Patel

(12MCEC42)

Guided By

Dr. Madhuri D. Bhavsar(Internal Guide)

Dr. Madhukar B. Potdar(External Guide)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2014

Certificate

This is to certify that the Major Project Report entitled ""**Efficient Live Virtual Machine Migration In Open Cloud Platform"** submitted by **Patel Pradip D.(Roll No: 12MCEC42)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Madhuri Bhavsar
Internal Guide & ,
Sr. Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.M.B.Potdar
External Guide &,
Project Director,
BISAG
Gandhinagar

Prof. Vijay Ukani
Associate Professor
Coordinator M.Tech - CSE
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Professor and H.O.D,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Undertaking for Originality of the Work

I **Pradip D. Patel**, Roll. No. **12MCEC42**, give undertaking that the Major Project entitled "**Efficient Live Virtual Machine Migration In Open Cloud Platform**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Madhuri D. Bhavsar

(Signature of Guide)

Acknowledgements

I would like to express my heartily gratitude to **Dr. Madhuri Bhavsar**,Institute of Technology, Nirma University, for giving me the valuable guidance to carried out the project work..

My deepest thank you is extended to **Dr. Madhukar B. Potdar and Mr. Miren Karamta**, Project Scientist, BISAG, Gandhinagar for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed whole heartedly to **Prof. Vijay Ukani**, Department of Computer Science and Engineering Nirma University,for giving constant encouragement.

I would also thank **Dr. Sanjay Garg**,Head of Computer Science and Engineering Department, Institute of Technology, Nirma University,for giving support and encouragement throughout the Project work.

A special thank you is expressed whole heartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, for the unmentionable motivation he has extended throughout course of this work.

I would also thank the all faculty members of Computer Engineering Department, Nirma University, and Ahmedabad for their special attention and suggestions towards the project.

Last but not list for most our sincere gratitude goes to the Almighty, the giver of life, health,wisdom and knowledge.

- Pradip Patel

12MCEC42

Abstract

Computing resources can be accessed on pay as you go basis in cloud computing. Virtualization is the core idea behind cloud computing. Live migration is the process of transfering a running virtual machine between different physical computer node without disconnecting the user.In live migration process system memory are transferred from the source physical host machine to the target machine. Live Migration is beneficial for load balancing and energy saving of physical machines in cloud computing.

Live migration leads to performance loss that cannot be ignored in cloud computing, if critical business goals are to be met. This work classify and evaluate current technique with respect to determine costs of virtual machine live migration with respect of total migration time.

The dissertation mainly focus on how migration process work and how to minimize the total migration time .Live migration problem is addressed by the various technique of migration.

Contents

Certificate iii

Undertaking iv

Acknowledgements v

Abstract vi

List of Figures ix

1 Introduction:Cloud Computing 1
- 1.1 What is Cloud Computing? . . . 1
- 1.2 Motivation . . . 2
- 1.3 Scope of the Work . . . 2
- 1.4 Organization of Thesis . . . 3

2 Purpose Of Virtual Machine Migration 4
- 2.1 Energy aware migration approach . . . 5
- 2.2 Load balancing migration approach . . . 5
- 2.3 Fault tolerant migration approach . . . 6

3 Literature Survey 8
- 3.1 Fundamental of Migration . . . 9
- 3.2 Pre-copy approach: . . . 9
  - 3.2.1 Start stage . . . 9
  - 3.2.2 Pre-Copy End stage . . . 10
  - 3.2.3 Stop-Copy stage . . . 10
- 3.3 Post-copy approach . . . 10
  - 3.3.1 Post Copy and its modified version . . . 10

4 The Proposed Algorithm 14
- 4.1 Design Metrics . . . 14
- 4.2 Migration Cost . . . 15
- 4.3 Migration Cost Parameter for live migration: . . . 15
- 4.4 Performance Model for Migration . . . 16
  - 4.4.1 Live Virtual Machine Migration Performance: . . . 19

5 Cloud Environment Setup: 23
5.1 Tool Used . . . 23
5.2 Open Stack Architecture . . . 24
5.3 Interface :Horizon) . . . 24
5.4 OpenStack Dashboard (Horizon) . . . 26
5.5 Services . . . 26
5.6 Flavors . . . 26
5.7 Network Configuration . . . 26

6 Result and Findings: 33
6.1 Configure Migration in Openstcak: . . . 33
6.1.1 Nova Environment: . . . 33
6.1.2 Configuration of system: . . . 34
6.2 After Succesful Migration . . . 34
6.3 Result and Findings . . . 35

7 Conclusion and Future Scope 38

8 List of Paper Published 39

List of Figures

1.1 Cloud Computing 2

2.1 Energy Efficient Migration Techniques [7] 5
2.2 Load balancing migration techniques [7] 6
2.3 Energy efficient migration techniques [7] 7

3.1 Live Migration algorithm [2] 9
3.2 Timeline for Pre-copy and Post-copy [2] 11
3.3 Pre-Copy Vs. Post-Copy Memory Transfer 12

4.1 Taxonomy of Migration Cost [2]. 16
4.2 Migration Cost Parameters [2] 17

5.1 Open Stack Architecture 24
5.2 Open Stack Module 25
5.3 Open Stack Instance communication flowchart] 26
5.4 OpenStack Dashboard Login 27
5.5 OpenStack Services 27
5.6 OpenStack Flavors 28
5.7 Open Stack Images 28
5.8 Open Stack Network Configuration 29
5.9 OpenStack Instances 29
5.10 OpenStack Instances Launch 30
5.11 OpenStack Instances Nettwork Configuration 30
5.12 OpenStack Instances Succesful Created 31
5.13 OpenStack Instances Succesful Running 31
5.14 OpenStack Instances Succesful Running in web-browser 32

6.1 OpenStack Instances Succesful Running in web-browser in destination host 35
6.2 Migration Time(S) of Different Vm in OpenStack Live Migration (Pre-copy) 36
6.3 Migration Time(S) of Different Vm in OpenStack Live Migration (Modified Pre-copy Algorithm) 37
6.4 Migration Time(S) of Different Vm in OpenStack Live Migration (Modified Pre-Copy Vs.Pre-copy) 37

Chapter 1

Introduction:Cloud Computing

1.1 What is Cloud Computing?

Cloud computing is almost each and every place. Visit any IT website or Pick up any magazine ,you all see "talk" about cloud computing. The main problem is that no one have the same opinion what actually it is. In year 2008 CEO of Oracle company examine cloud computing issue and saying that the term was overused in the computer world.Cloud Computing defines by Berkeley RAD Lab as follows: Cloud Computing include both the hardware and systems software in the cloud computing and the applications delivered as services . The services is known as Software as a Service.In public cloud ,cloud is made available in a pay-as-you-go manner to the public;while private cloud is only for particular oraganization and not available to the public.
Thus, Cloud Computing is the combination of Utility Computing and software as a service, but does not include Private Clouds.

Cloud Computing term has been explain in number of way by professional it companies,academics, industry practitioners, analyst firms.Cloud computing is a large pool of accessibleand usable resources like hardware, development platforms and services.Resources can be runtime configured to adjust workload , allowing for an optimum resource utilization also called load balancing.

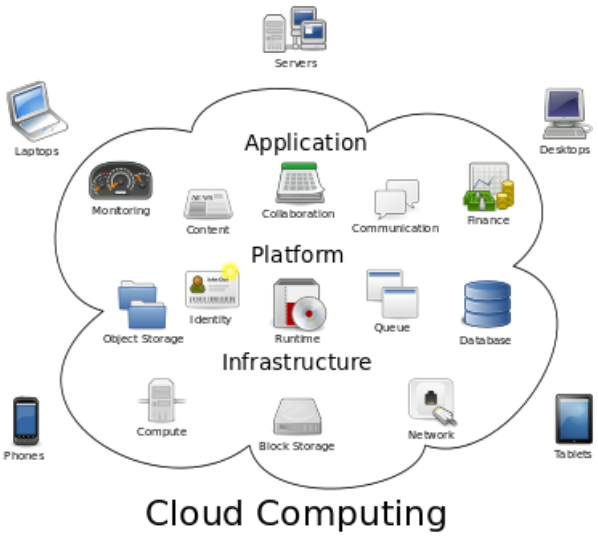

Figure 1.1: Cloud Computing

1.2 Motivation

Virtual Machine are placed in servers so that clients requests are serviced in an best possible manner.Virtualization is the main key concept of cloud.Live migration is the process of transfering a active virtual machine from one host to another without disconnecting the client in cloud. Live Migration is basically for getting energy efficiency,high availability of physical machine and for load balancing.

1.3 Scope of the Work

Proposing a taxonomy for live migration,a migration costs and the parameters that may influence migration process.Based on previous work which is already done for live virtual machine migration evaluating previous results of migration.And resume with which migration techniques were already examined, which parameter they depends and how it can be expressed in formal models.Identifying and evaluating open issues and propose important parameter that have not bear in mind in previous work.

1.4 Organization of Thesis

Following this introduction, chapter 2 presents an overview of the traditional migration techniques, followed by introduction of migration and its requirements.

Chapter 3 presents a literature survey of previous work to date in the domains of Live Migration that is related to the work presented in the remainder of the thesis.

Chapter 4 proposes an algorithm based on pre-copy memory migration technique with the detailed explanation of the algorithm.

Chapter 5 describes the methodology used in implementing each step of the algorithm whereas chapter 6 shows the experimental results .

The thesis ends with discussing the conclusions derived from the work and explores some future enhancements that can be made to the algorithm in Chapter 7.

Chapter 2

Purpose Of Virtual Machine Migration

In year 2008 Google estimated 200,000 to 500,000 servers, included in 36 data centers. Microsoft Added 10,000 servers per month in 2008 FaceBook added More than 30,000 servers in its data center in 2008 .Cloud Infrastructures and data centers have to efficiently use their huge scales of hardware resources.

Roles of a Data Center: (1)Flexibly remap hardware among VMs. (2)Balance workload (3)Save energy (4)Enhance service availability and fault tolerance.

Live migration is a process that migrate the whole OS and its application from source physical host Machine to destination machine.The Virtual machines are migrated lively Without disconnecting the application running on source physical machine. The advantage of live virtual machine migration include energy overheay, fault tolerance and load balancing.

Virtual machine migration approach listed below:

2.1 Energy aware migration approach

The power eating of cloud data server center is mainly based on the utilization of the data servers and their cooling systems. The Data servers need up to 75consumption even at their idele condition.So there is a need for migration techniques that save the power of servers by optimize the resource utilization at their pick level[7].

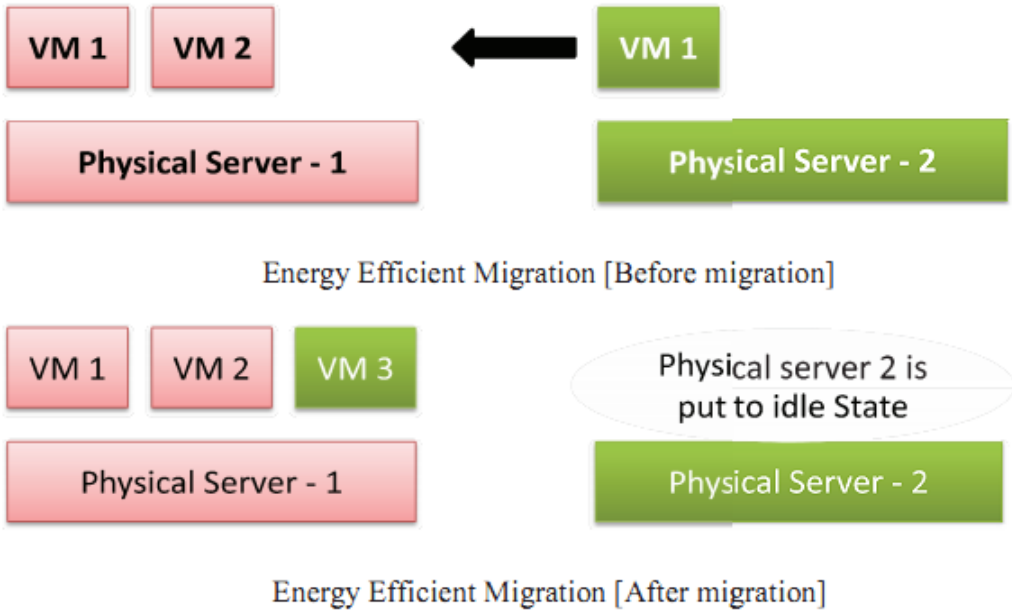

Figure 2.1: Energy Efficient Migration Techniques [7]

2.2 Load balancing migration approach

The Load balancing migration approach basically for distribute load across the physical data servers to improve the scalability of physical data servers in cloud computing firm. The Load balancing focus in minimizing resource consumption, implementation of fail-over technique,enhancing scalability of cloud and avoiding over-provisioning and under-provisioning of cloud resources etc[7].

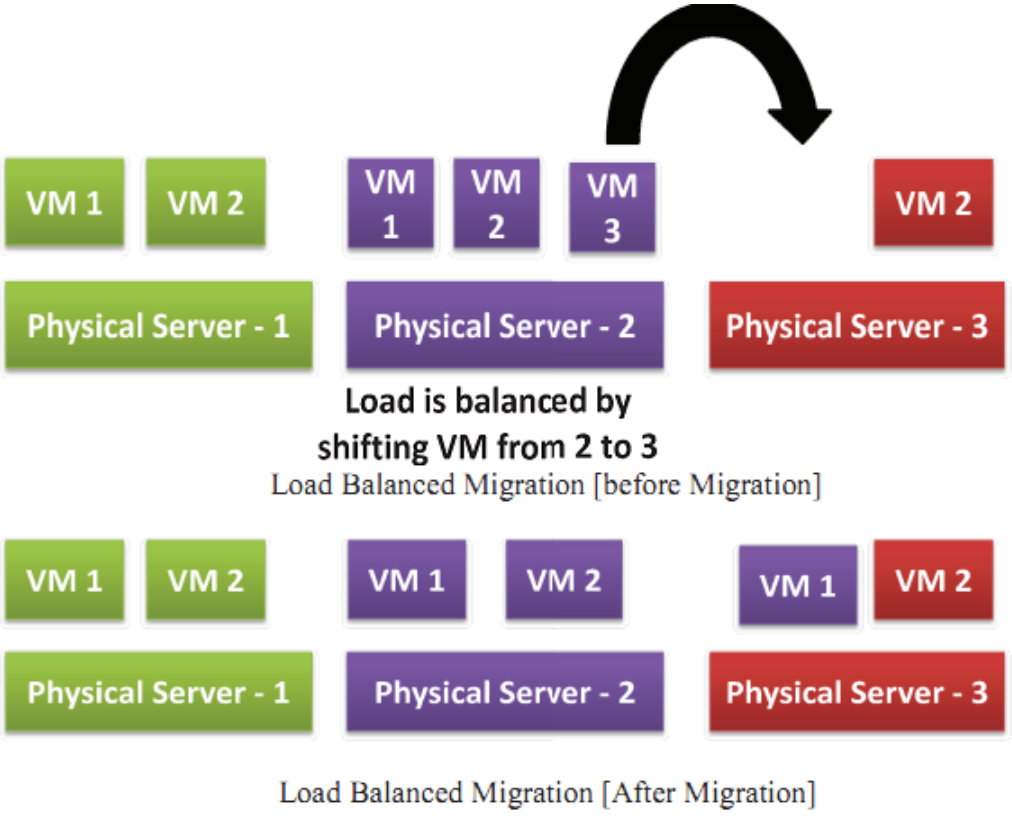

Figure 2.2: Load balancing migration techniques [7]

2.3 Fault tolerant migration approach

Fault tolerance migration approach allows the physical machines to continue its work if any part of the machine fails. This approach migrates the vm from a one physical machine to the another physical machine based on the future predication of the failure.The advantage of fault tolerant migration approach is to get better the availability of physical machine and prevent performance humillation of application which is running on it[7].

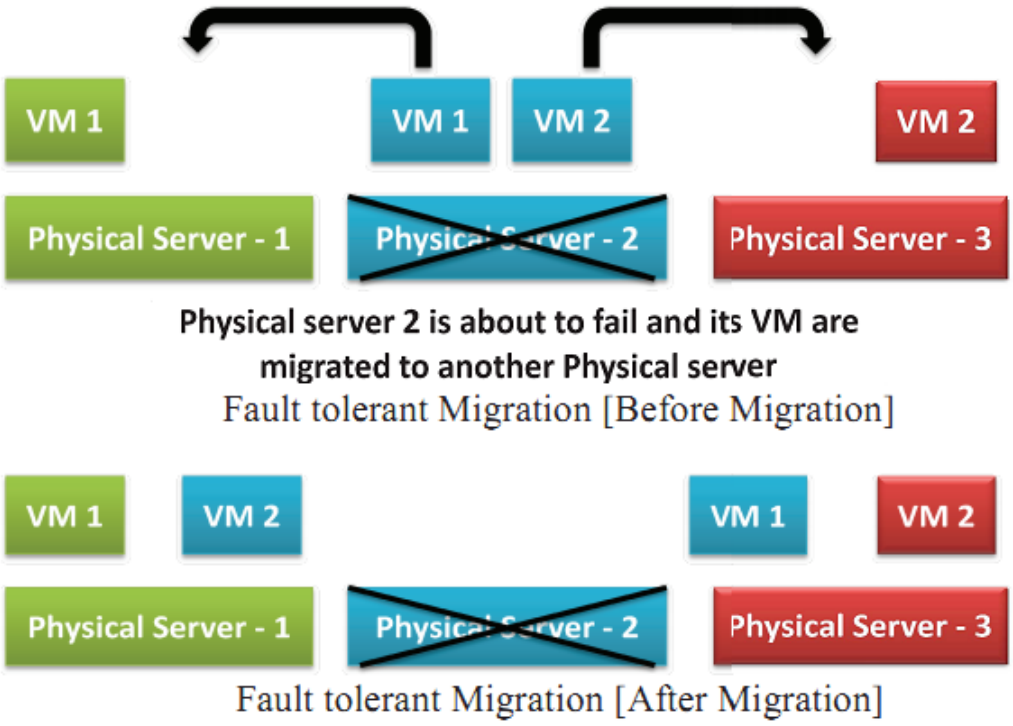

Figure 2.3: Energy efficient migration techniques [7]

Chapter 3

Literature Survey

Virtualization facility make allows more then one operating systems run simultaneously on the same physical host machine. Virtualization technology give the facility to migrate vm from one source machine to another destination machine.

Virtual Machine Migration is a very useful for admin of Cloud data server center: it allows clean division between software and hardware.Live Migration is beneficial for efficient resources utilization , load balancing, energy saving.

Vm Migration methods are mainly two types:(1) live migration also refer as hot migration. (2) non-live migration also refer as cold migration

Live migration:

Vm continue running while migrating process start and does not lose its status.client doesnt feel any interruption in service in live migration.

Non-live migration:
status of the VM loses and user can notice the service interruption in non-live migration.

In live migration process, the state of a virtual machine to migrate is transferred. The state consists of its memory pages and local file system. Local file system need not be transferred. First, VM is suspended, then its state is transferred, and lastly, VM is

resumed at destination host.[1].

3.1 Fundamental of Migration

In Live migration process, virtual machines transfere from one physical machine to another physical machine, while the virtual machines are on. The process of live migration is limited to copying content of CPU registers and the memory state in between the hypervisors of cloud computing application software. To ensure that, modern cloud computing systems use two methods (1)pre-copy and (2)post-copy technique .

3.2 Pre-copy approach:

Pre-copy approach,divided into three parts as shown in figure 3.1:

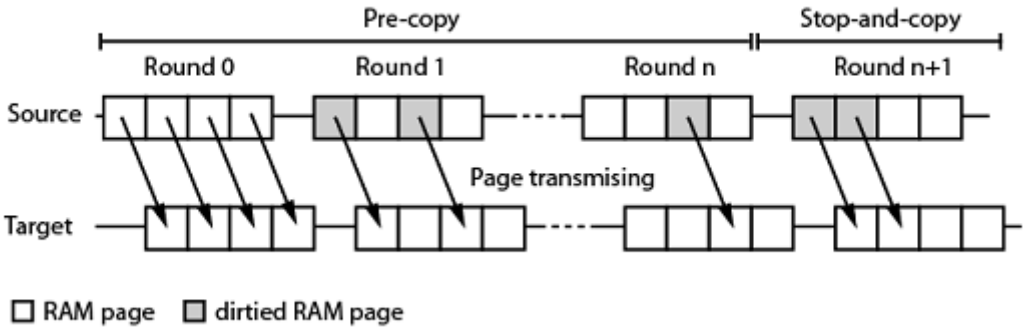

Figure 3.1: Live Migration algorithm [2]

3.2.1 Start stage

Start stage: In this stage, the Virtual Machine is continues run on source machine, while virtual machine's memory pages is iteratively copy from the source machine to the target machine. It need some time to copy memory pages and in between the Virtual Machine is runing on the source machine, some pages may be change by user ,that pages called dirty pages and have to be resend in an next round to prevent memory consistency of the system.

3.2.2 Pre-Copy End stage

Pre-Copy End stage: Pre-copy end stage depend on hypervisor of cloud platform software that used threshold of iterations of memory pages already transfered.

3.2.3 Stop-Copy stage

Stop-Copy stage: In this level , hypervisor of cloud platform software suspends the Virtual Machine to stop the process of changing the page which is called dirty pages and copy the remaining dirty pages and processor registers status to the destintion machine.At the end ,hypervisor of cloud computing platform software resumes the Virtual Machine.

3.3 Post-copy approach

In this approach hypervisor of cloud platform software first suspends the migrating Virtual machine at the source physical machine, copy minimum cpu state to the target machine and resumes the vm at target node and then start fetching memory pages from the source machine.

3.3.1 Post Copy and its modified version

Post Copy technique with Demand Paging: Post Copy technique with Demand Paging is very simple .When Virtual Machine resumes at the destination host, the memory accesses for page faults serviced by requesting the referenced page from the source machine. So, servicing every page-fault will slow down the Virtual Machine because of round trip time. Each page is transferred only once. So, post copy technique performance for this approach by itself would be unacceptable regarding migration time.

Post Copy technique with Active Pushing:

In this technique the Virtual Machine pages is push from the source machine to the destination machine although Virtual Machine running on the destination machine.If page fault is occured then Virtual Machine can be serviced concurrently with the help of demand paging. Active pushing technique avoids transferring fault pages to be transfered by the target Virtual Machine.So, each memory page is transferred by only once with the help of active push.

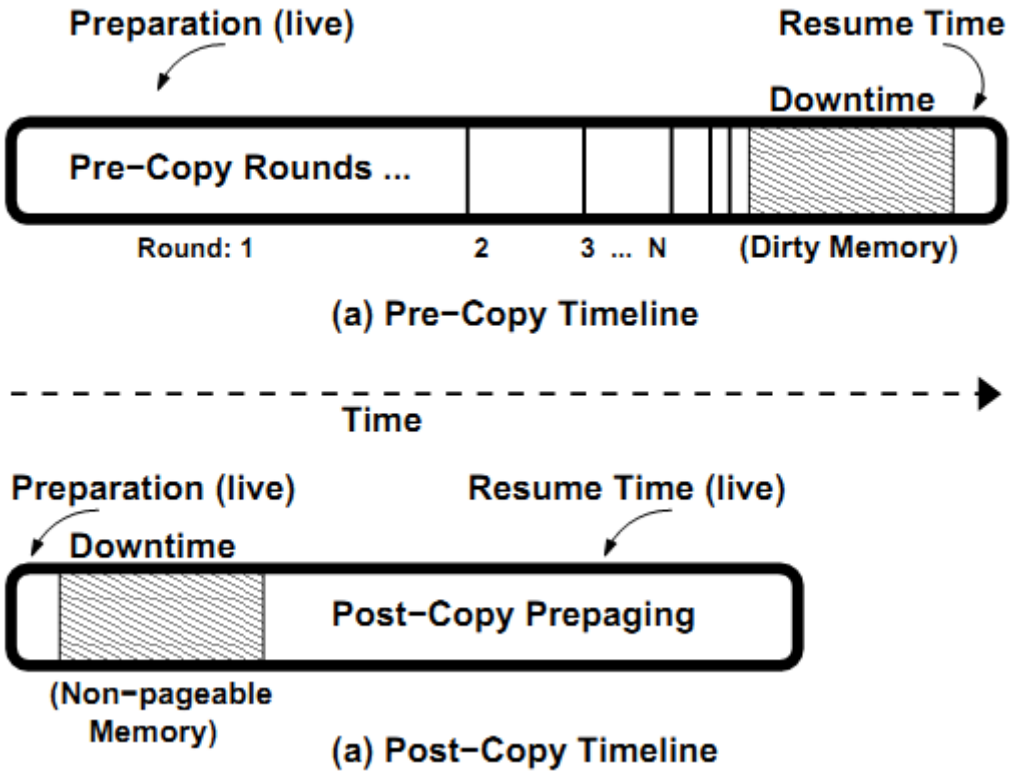

Figure 3.2: Timeline for Pre-copy and Post-copy [2]

Post Copy technique with Prepaging: The main fundamental behind post copy withthe help of prepaging is to anticipate the chance of major faults to be occure in advance and adapting the memory page pushing sequence to better reflect the Virtual Machines memory access pattern. It is not possible to predicting the when page fault of Virtual Machines occure, So technique works by with the help of the page fault address as hints to finalized the locality of the Virtual Machines memory access pattern.

Source:http://grivon.apgrid.org/quick-kvm-migration

The basic idea of live migration algorithm, first proposed by Clark [5]. First Hypervisor marks all memory pages of source node as dirty pages, then transfer dirty pages in several round to the destination node and stop this process when number of pages remaining to be transferred from source machine to destination machine is less then predefined threshold of maximum number of round is reached. Then Hypervisor mark transferred pages as clean, So there is chance to pages become dirty which is already transferreed and there is need to be retransfer that pages also. The Virtual Machine is stop at some point on the source machine node for stop further memory pages writes and transfer remaining

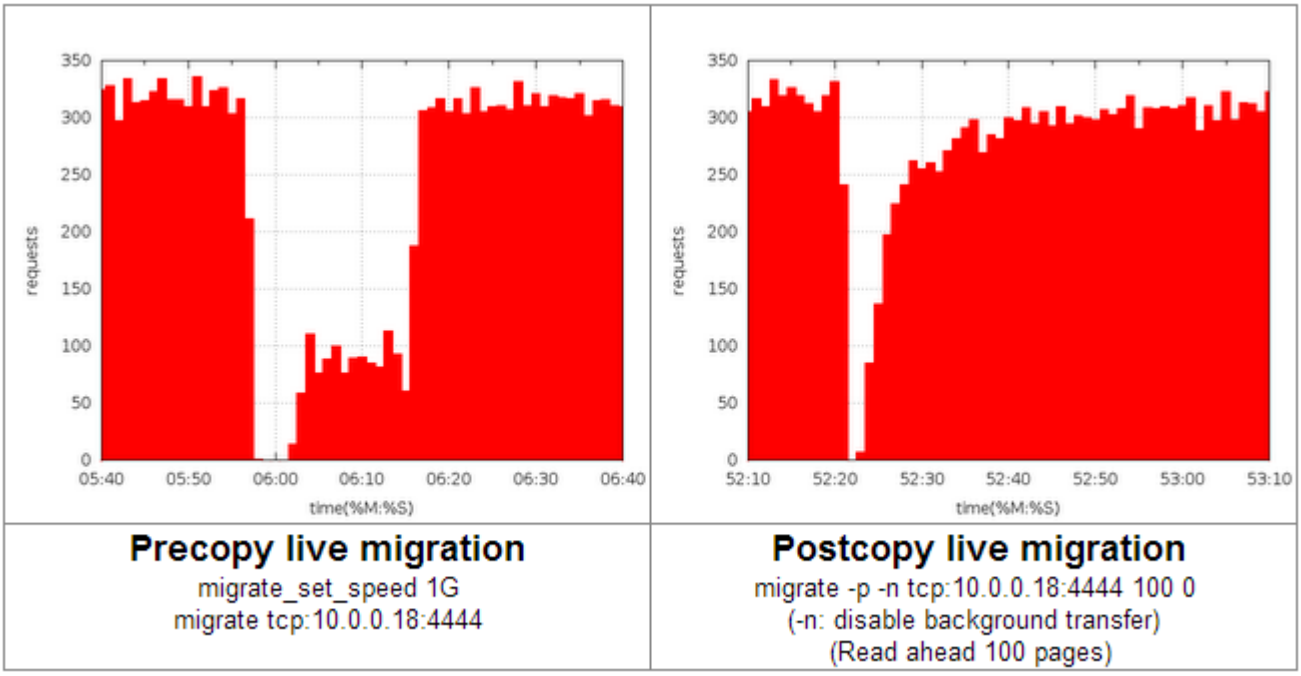

Figure 3.3: Pre-Copy Vs. Post-Copy Memory Transfer

pages which is dirty pages.After transferring all the memory contents, the VM resumes at destination..

The implementation describes by Nelson [6] of a migration process that uses vm technology to provide fast application migration, neither any of applications or the os need to be modified. Performance is measured with hundred virtual machines, migrating concurrently with standard industry benchmarks. It shows that for a variety of workloads, application downtime due to migration is less than a second.

Remote Direct Memory Access (RDMA) was proposed by Huang et al. [7]. InfiniBand is an emerging interconnects that give high performance and features such as Operating System bypass and RDMA. RDMA is a direct memory access from the memory of one source machine into that of another destination machine without help of source and destination machine os. By using RDMA remote memory can be read and write (modified) directly, hardware I/O devices can directly access memory without involving OS.

Luo [8] present a technique,in which transfers the virtual machine run time state, including processor state, memory pages, and disk storage.He describe a three phase mi-

gration algorithm and incremental migration algorithm technique, which migrate the vm back to the source node in a less time. During the migration process, all the write operation of memory pages to the disk storage are monitoring with the help of Block bitmap technique.

Kuno, et. al.[2] proposed technique for performance evaluation of live and non-live migration methods and show that performance of apllicaction on a migrating vm severely declines. The important reasons for the decline are a source machine and destination machine Operating System . They also analyze the reasons of I/O performance decline.

Feng et. al. [9] describe performance of VMotion . VMotion performs good in generating total virtual machine migration data when migrating Virtual Machine instance than XenMotion. The performance of VMotion degrades with network delay and packet loss of data. VMotion performs worse in the presence network delay and packet loss of data.

Chapter 4

The Proposed Algorithm

The behaviour of **live and non-live Virtual Machine migration** technique could be evaluate with the help of listed below metrics [1].

4.1 Design Metrics

- **Preparation Time**:

  The time between initiating virtual machine migration process start and transferring the Virtual Machines cpu state to the destination machine,during which the Virtual Machine continues to run.

- **Down Time** :

  The time in which Virtual Machines execution is stopped known as down time .

- **Resume Time**:

  Require time for resuming the Virtual machineMs execution at the destination and the end of migrational together is knows as resume time at that stage all dependencies on the source node must be eliminated .

- **Total Pages Transferred**:

  Total count of number of memory pages and number of dirty page transferred in above time periods. post copy approach transfers most during resume time and Pre

copy approach transfers most of memory pages during preparation time.

- **Total Migration Time** :
  The time between star of migration and stop the migration process is known as total migration time. Total migration time is very crucial because it affects the release of resources off source machine and destination machine.

4.2 Migration Cost

virtual machine migration process performance is important aspects.Two parameters is very crusial: (1)The total migration time and
(2)The down time.

Machine's performance loss is another important parameter during live migration process. To identify dirty pages, the source machine hypervisor of cloud computing platform software has to observe Virtual Machines memory, which slows down migration process because of write operation of memory and thus execution and throughput of application running on the Virtual Machine[9].

In pre copy start stage and stop-and-copy stage, the hypervisor of cloud computing platform software requires extra resources like network input and output module and processor cycles. If there are not enough resources available, the resource allocation of the migrated Virtual Machine is reduced and hence performance loos occurs.

another important parameter is energy overhead of live migration. It is due by the network interface and resource utilization like processor.Live migrations energy overhead to be considered, in the case when migration is supposed to optimize power consumption in cloud data server centers.

4.3 Migration Cost Parameter for live migration:

live virtual machine migration has two main goals:

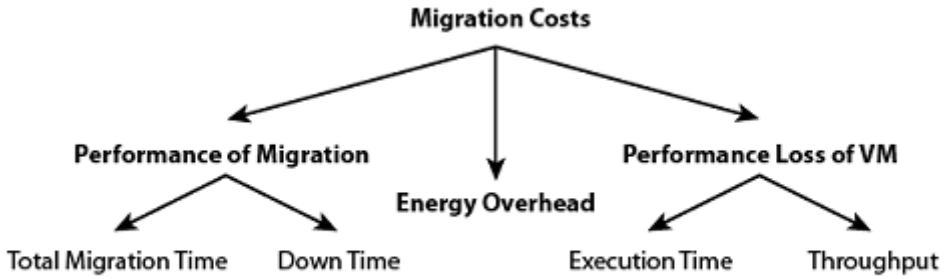

Figure 4.1: Taxonomy of Migration Cost [2].

(1) important parameters depends on which things.(2) design live migration models that predict migration costs based on that parameters.

Mainly three parameters influence migration costs:

(1) Parameters of physical node.(2) Parameters of vm and (3) Parameters of infrastructure ussed for live migration process .

Parameters of physical node refer to the resource utilization of source machine and destination machine.Beacuse live virtual machine migration generate extra load on processor and on nic cards.Performance and energy consumption of physical machine depend on the physical machine's current load , main memory and local hard disc.

Parameters which is related to Virtual Machine show the behaviour of the Virtual Machine that migrated from source machine to destination machine, Because time taken by pre-copy approach highly depends on virtual machine memory size and on how many pages become dirty page ,So add the size of the Virtual Machines memory and page dirty rate to design live migration model.

4.4 Performance Model for Migration

Symbol :

t_{mig} = Total Migration Time[second].

t_{down} = Down Time in [second].

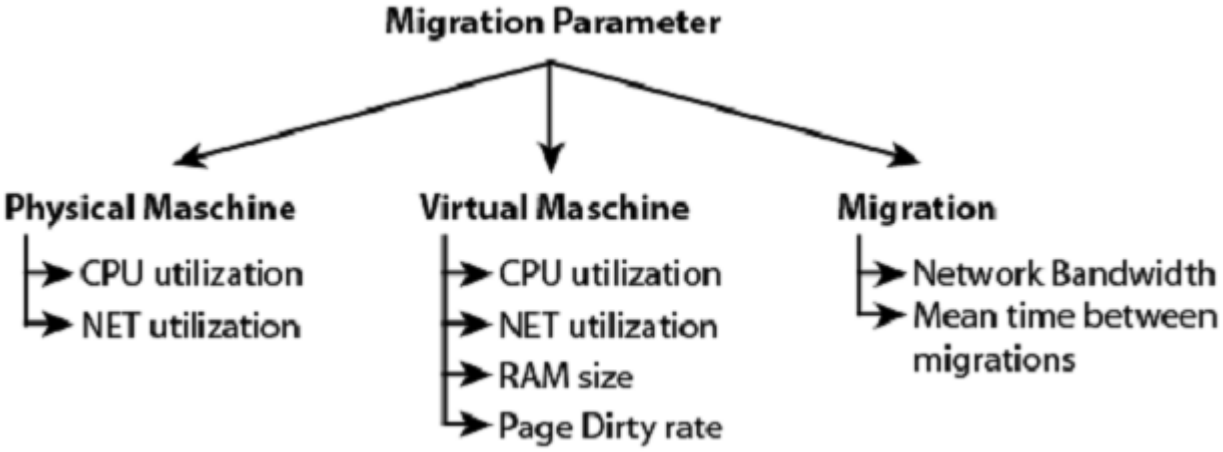

Figure 4.2: Migration Cost Parameters [2]

v_{mem} = Virtual Machines memory size.

d = dirty page rate [page/s].

b = network bandwidth between source machine and destination machine.

Algorithm : Proposed Live VM Migration Algorithm:-

/* **Input** */ n_{th} ,v_{th} ,p_{th} ,v_{mem} ,b ,d ,l

/* **Output** */ t_{mig} ,t_{down} //

/* **Initialization** */

$v_0 \leftarrow \mathrm{v}_{mem}$

$v_{mig} \leftarrow 0$

$\mathrm{t}_{mig} \leftarrow 0$

$\mathrm{t}_{down} \leftarrow 0$

/* **Main loop** */

For i = 1 to m_{th} **do**

$t_i \leftarrow (\mathrm{v}_i)/b$

$v_{i+1} \leftarrow$ d * l

$\mathrm{t}_{mig} \leftarrow \mathrm{t}_{mig} + t_i$

$v_{mig} \leftarrow \mathrm{v}_{mig} + v_i$

If $v_{mig} > v_{th}$ **then**

break

End If

End For

$t_{down} \leftarrow \mathrm{v}_{i+1}/b$

$t_{mig} \leftarrow \mathrm{t}_{mig} + t_{down}$

Virtual Machine size and memory page size and network bandwidth are already known, As the memory page dirty rate changes respect to time. and intially dirty pages is approximately the same at the virtual machine workload.

4.4.1 Live Virtual Machine Migration Performance:

1. **Calculating Migration Time**:

   The migration time t_{mig} depends on,

   (1) total amount of virtual machine memory v_{mig} that is transmitted from source machine to target machine

   (2) network bandwidth b between source and target machine [6].

   So ,it can be

$$t_{mig} = \frac{v_{mig}}{b} \tag{4.1}$$

   Live virtual machine migration process copy vm memory in many rounds.Intially, it copy entire RAM from the source machine to the destination machine,and then resend the pages that have been dirty.

   Virtual machine memory pages that may be transfered, depends on the Virtual Machines memory pattern and page dirty rate d , [8].

   The dirty rate of memory page calculated by how many pages are written within on second in memory.

   Let l be the page size of memory and t_{i-1} is the time taken by previous round, Liu et al. [9] calculate the amount of memory $v_{mig,i)}$ to be transferred at round i from source node to destination node as in

$$v_{mig,i} = \begin{cases} v_{mem} & i = 0 \\ d * l & else \end{cases} \tag{4.2}$$

Assuming the live virtual machine migration process uses n rounds plus the final stop-and-copy stage round, the total memory v_{mig} is

$$v_{mig} = \sum_{1}^{n} v_{mig,i} \tag{4.3}$$

Based on the equations in (4.1), in (4.2) and in (4.3) estimate the upper bounds and lower bounds of migration time for live migration process[6].

In the senario of idel condition ,an idle Virtual Machine runs no applications and not modify its memory content. So, the source hypervisor of cloud computing platform software sends the total memory record in the first round of pre-copy round and finishes the migration process.

In another senario,Virtual Machine whose dirty rate is high ,in this case each pre copy round including the first one, consider all memory pages are being modified. So, the live virtual machine migration system would have to transfer the whole memory in each round of pre-copy. At the last pre-copy phase stop if the Virtual machines whole memory is transferred m_{th} times

2. **Calculating down time:**

The live virtual machine migration process suspends the Virtual machine during the final round and copy the remaining dirty memory pages to the destination machine.

So,After some time Virtual Machine continues to run on target host. As per page dirty rate d, the page size l, the link speed b , the the down time t_{down} as in

$$t_{down} = \frac{d * l * t_n}{b} \tag{4.4}$$

The Upper limit and Lower limitof down time highly depend on the dirty rate, the live virtual machine migration algorithm copy the whole Virtual Machine memory , as in

$$0 \leq t_{down} \leq \frac{v_{mem}}{b} \tag{4.5}$$

Lower bound limit reached, if the Virtual Machine is set idle and not change any memory page, the upper bound limit reached if the page dirty rate is above bandwidth of network.

3. **Estimation of Live Virtual Machine migration Algorithm:**

As per shown in algorithm migration time and down time is parameterized with the help of following stop conditions:

(1) The maximum number of rounds n_{th},
(2) Total amount of memory $v_{[}th$ and
(3) the minimum number of dirty pages p_{th} in last round.

And the Virtual Memory size v_{mem}, bandwidth b,page dirty rate d and the page

size l has to be known.

The live vm migration process runs maximum m_{th} rounds.In every round the total memory to be copy is calculated by quation (4.1) and (4.2). In each round the amount of memory and the number of dirty pages are compared with thresholds value.

The live virtual machine migration process terminate if any of these thresholds is exceeded and algorithm give the migration time and down time.

Chapter 5

Cloud Environment Setup:

This chapter covers the details of the tool used for implementation of the project and other relevant details.Also explains the significance of some steps on the final output. In this section implementation strategy is given with the computation expression and on which platform that can be implemented and analyzed.
Openstack cloud environment is used to make private cloud, into that with the use of virtualization components of openstack like
(1)nova-compute, (2)nova-scheduler

VM are created as per the request of user. This report is based on the live virtual machine migration So, it need to implement proposed algorithm in with the use of nova-api using python scripting language.

5.1 Tool Used

OpenStack is a collection of open source software that facilate to setup cloud. Rackspace and NASA are contributors of the openstack. Rackspace contributed their "Cloud Files" platform to power the Object Storage module of the OpenStack and NASA contributed their Nebula platform to power the Compute part.

The experiments are set up on three physical machine.Each machine has a 2.13 Ghz Intel core i3 processor, 4 GB Ram,320 Gb Sata Harddisk and Ubuntu 12.04 with Linux

kernel(with KVM module) and Openstack Cloud.

Simulation results may be vary, if there is change in configuration of physical machine.

OpenStack have five main module namely nova swift,glance,keystone,horizon.

5.2 Open Stack Architecture

Openstack works with many components to do computation, network- ing, image management, storage management, process management, security managemenet using Identity componenet etcs. All the different componenets are discussed in this section

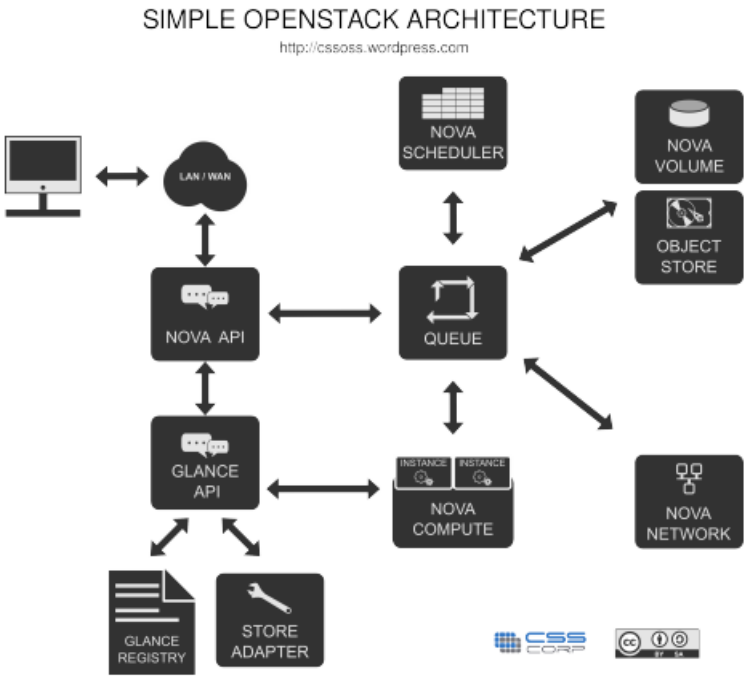

Figure 5.1: Open Stack Architecture

The listed below figure shows a representation of communication between keystone and dashboard with the OpenStack main module.

5.3 Interface :Horizon)

Horizon is a dashboard that can be used to manage OpenStack cloud services. That can be used to create/manage image and instance, create keypairs for cloud, attach storage to instances etc. also dashboard gives the user access to instance console and connect to

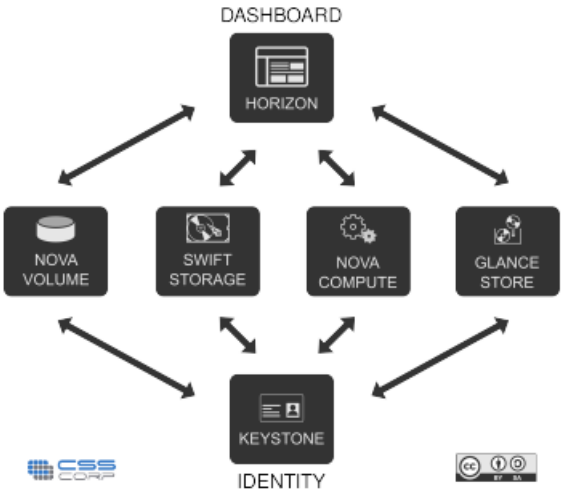

Figure 5.2: Open Stack Module

an instance through VNC client.

Following package is installed before going further. Configuration of nodes:(for 3 nodes)

- Ubuntu 12.04 LTS
- OpenStack Essex
- KVM And vmbuilder
- MySQL
- Identity Service (Keystone)
- Identity Service
- Image Service (Glance).
- Compute (Nova).
- Add images.
- OpenStack Object Storage (Swift).
- OpenStack Dashboard

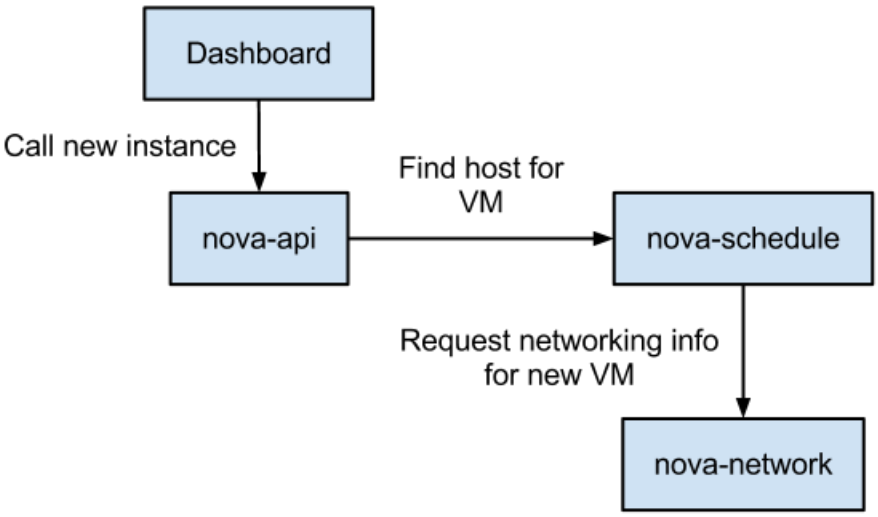

Figure 5.3: Open Stack Instance communication flowchart]

5.4 OpenStack Dashboard (Horizon)

Login to the dashboard with username ”admin” and password ”cloud”

5.5 Services

The list of services defiend can be viewed on this page.

5.6 Flavors

flavors module used to launch an instance with custom configuration as per requirements.

5.7 Network Configuration

On This page configure the network for cloud setup.

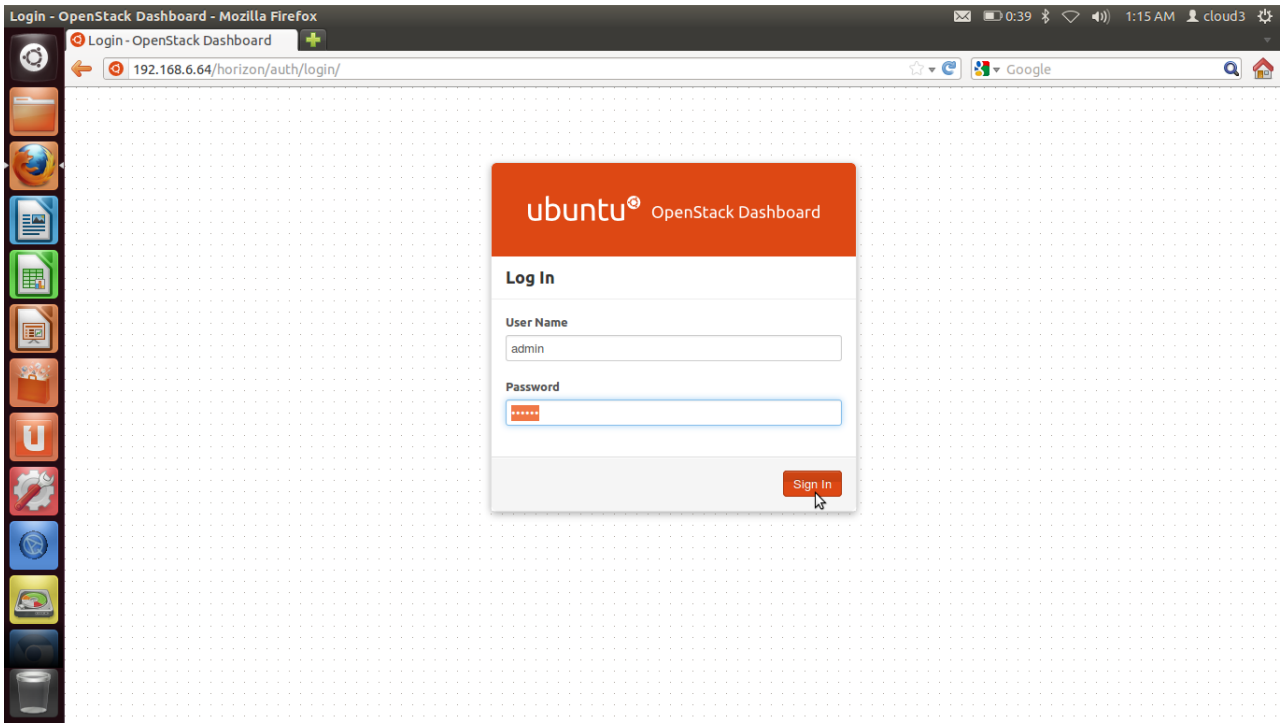

Figure 5.4: OpenStack Dashboard Login

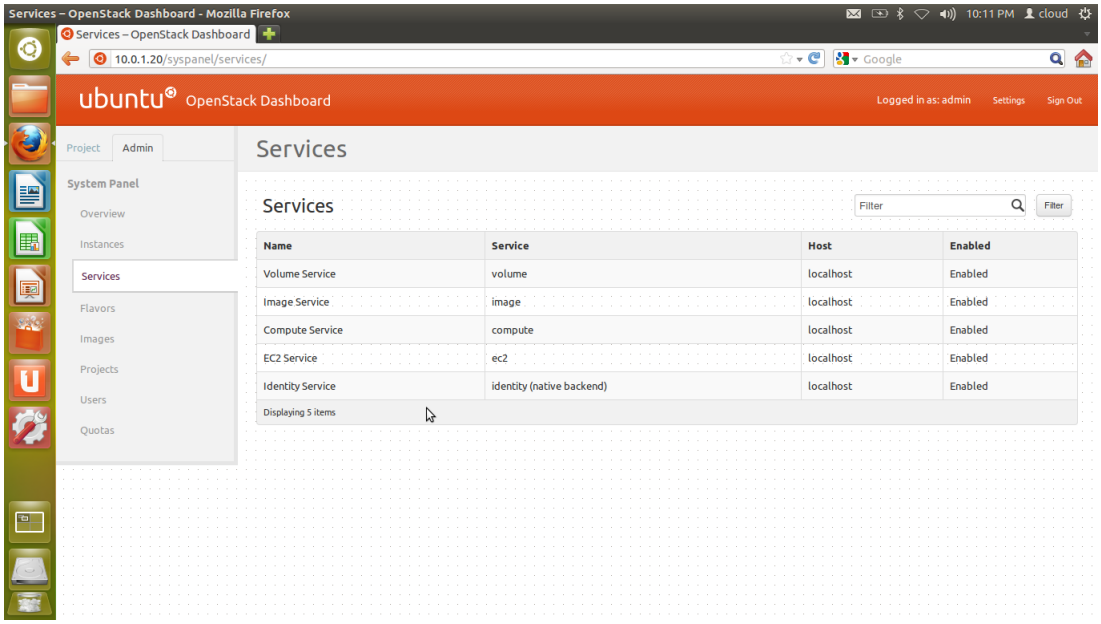

Figure 5.5: OpenStack Services

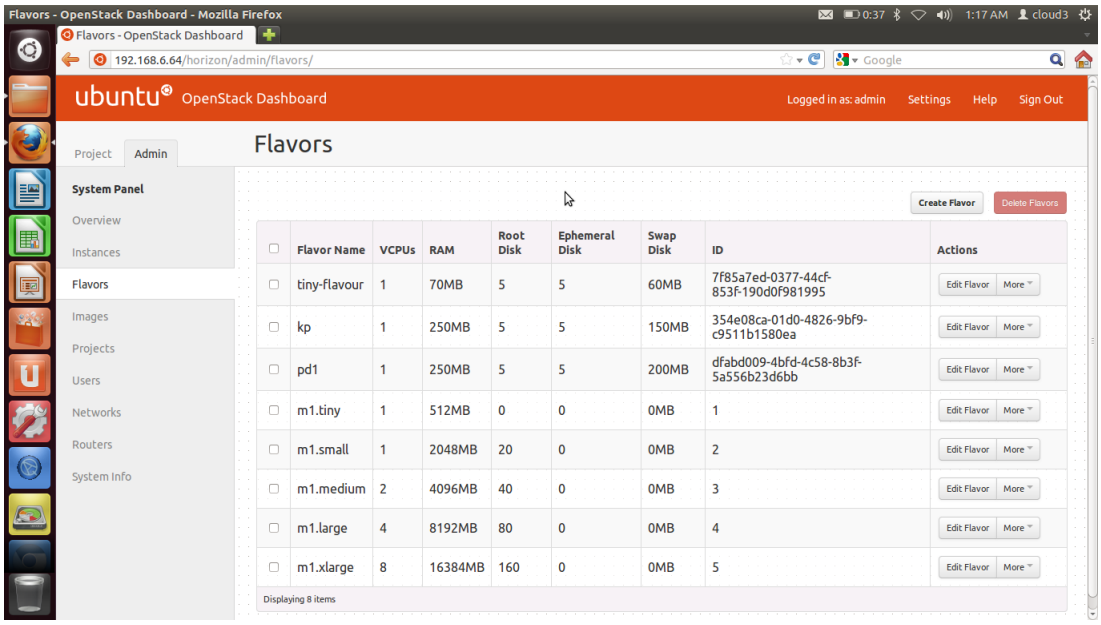

Figure 5.6: OpenStack Flavors

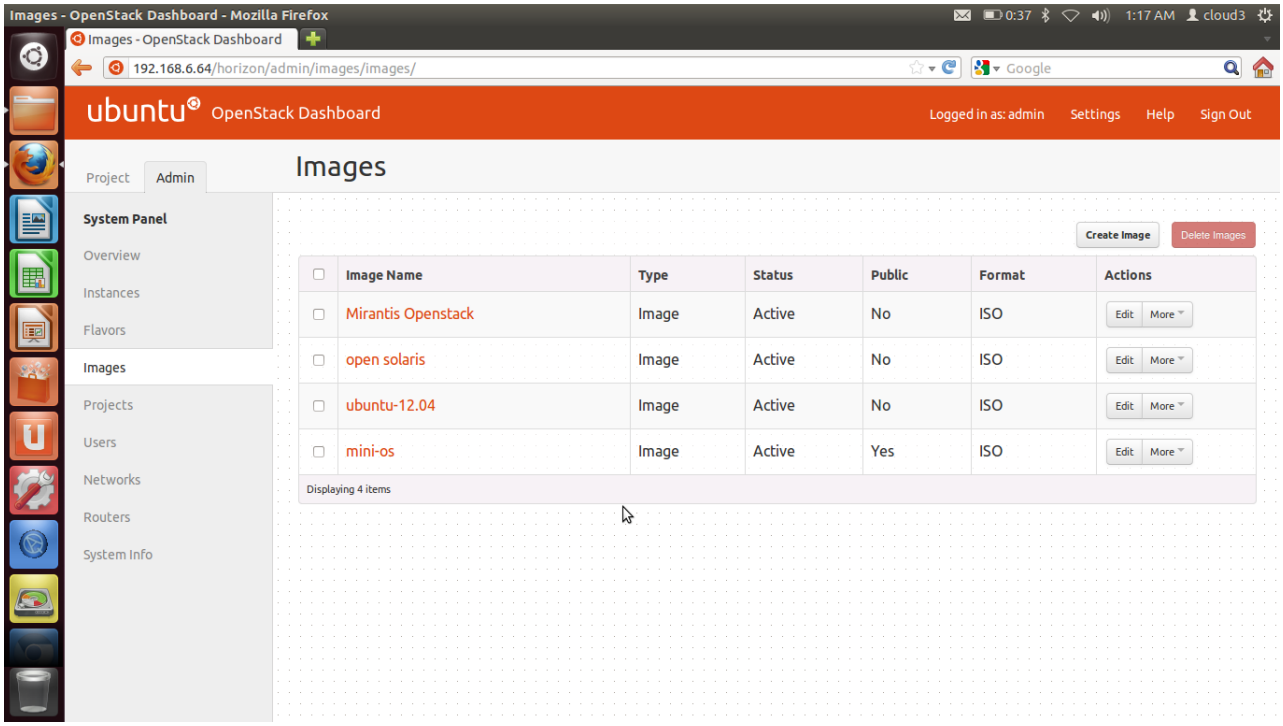

Figure 5.7: Open Stack Images

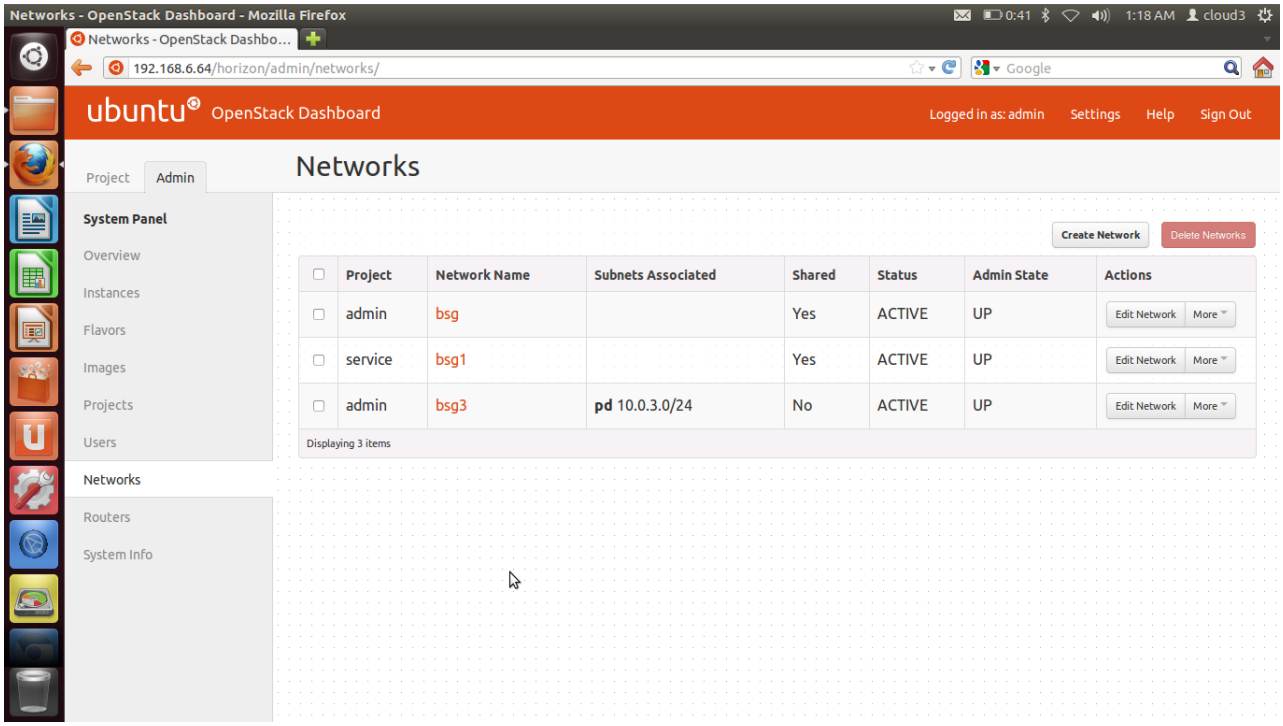

Figure 5.8: Open Stack Network Configuration

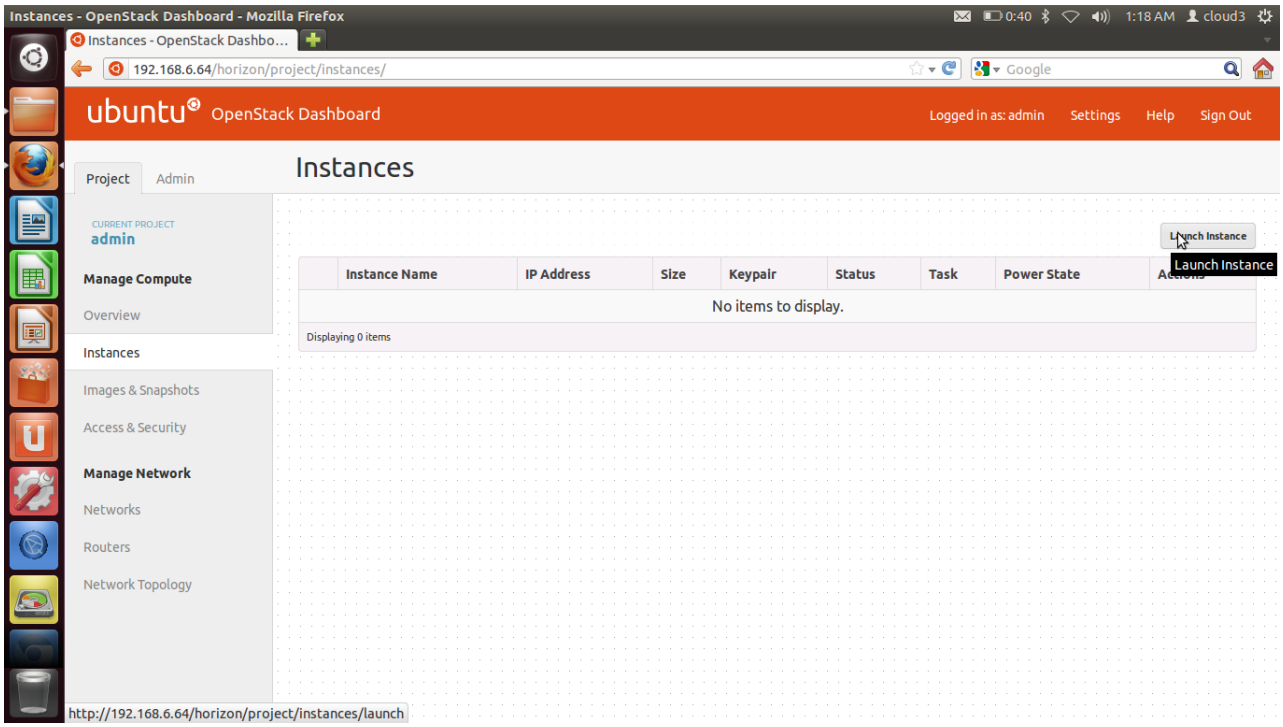

Figure 5.9: OpenStack Instances

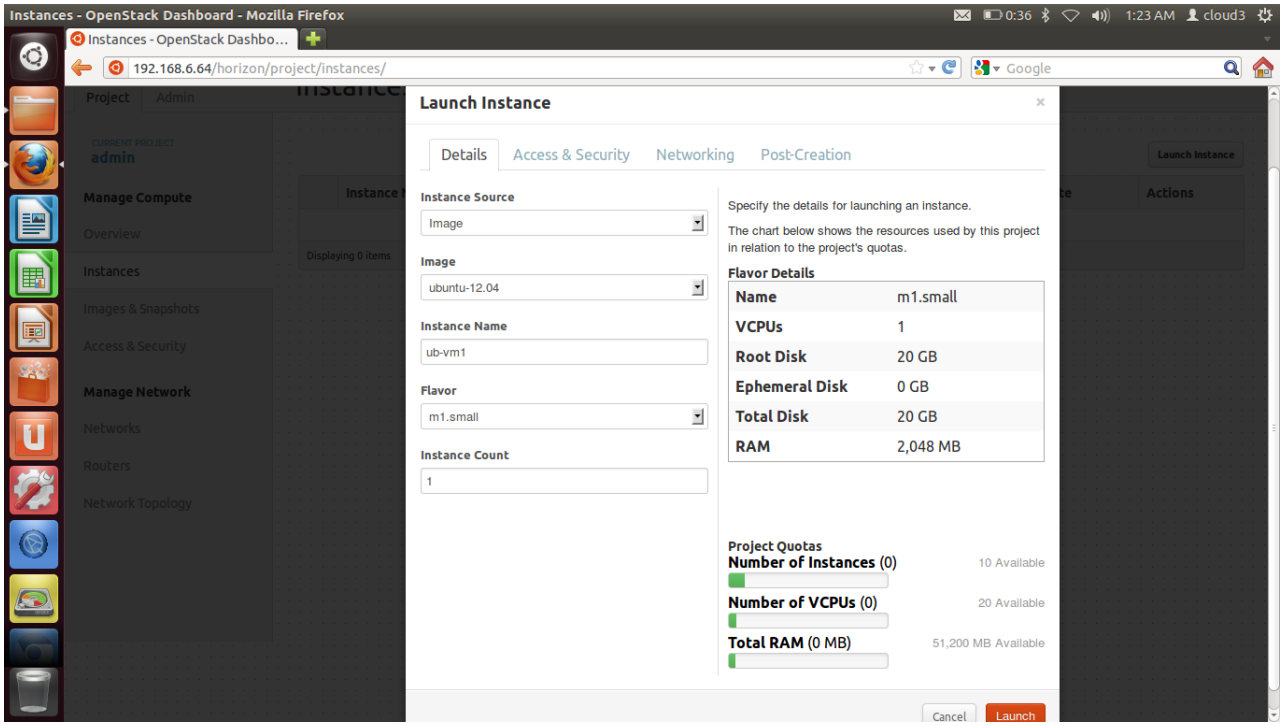

Figure 5.10: OpenStack Instances Launch

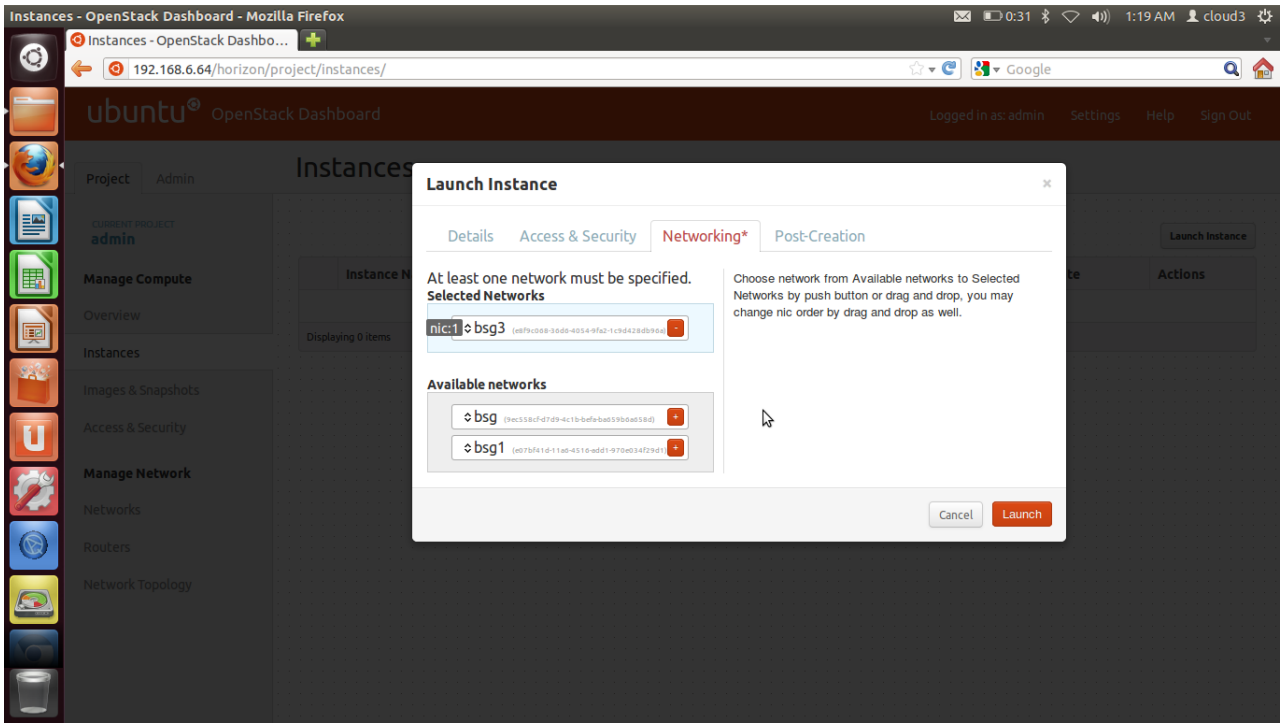

Figure 5.11: OpenStack Instances Nettwork Configuration

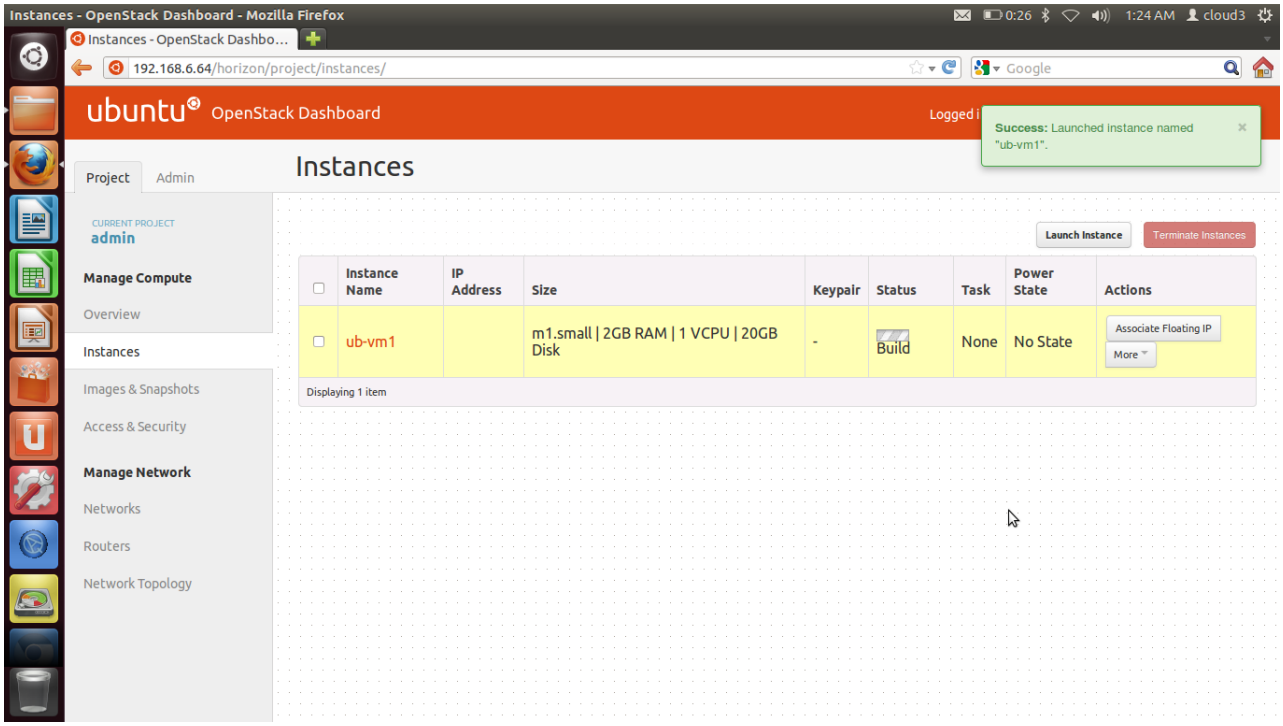

Figure 5.12: OpenStack Instances Succesful Created

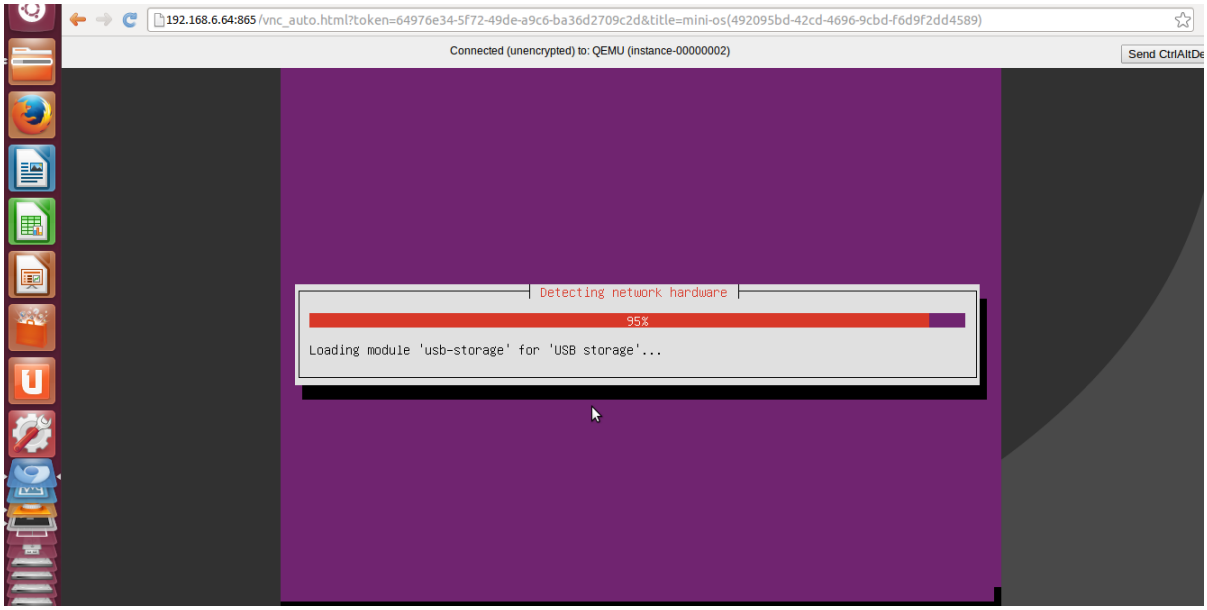

Figure 5.13: OpenStack Instances Succesful Running

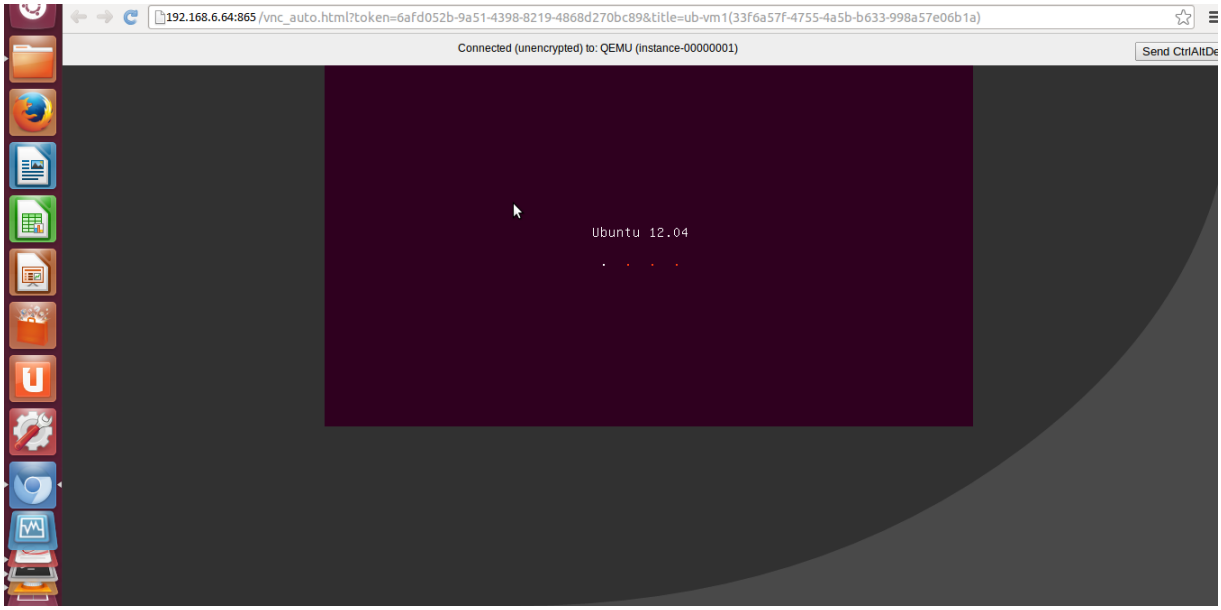

Figure 5.14: OpenStack Instances Succesful Running in web-browser

Chapter 6

Result and Findings:

6.1 Configure Migration in Openstcak:

Live Migration of virtual machine allows an admin to move a virtual machine instance from one source node to another destination node.
This service is very useful when a compute node requires maintenance and also some major advantages discuss earlier. Migration can also be useful to balancing the load when thousand of Virtual Machine instances are running on a cloud server.

6.1.1 Nova Environment:

Prepareing 3 node; the ip address is PC 1 :192.168.6.64 ,
Pc 2 :192.168.6.65, Pc 3 :192.168.6.66

PC 1 is the "Cloud Controller", and should running: nova-api, nova-scheduler, nova-network, cinder-volume, nova-objectstore.

nova-compute is running PC 2 and PC 3 called compute node.

nova.conf is same in all node.

PC 1 is the server which exports instances, and PC 2 and PC 3 mount it.

6.1.2 Configuration of system:

Configure DNS and check it is same in all nodes.
Use ping command to perform name resolution with each other.

Ensure that the UID and GID of your nova and libvirt users are identical between each of your servers. This ensures that the permissions on the NFS mount will work correctly.

Configure NFS at HostB and HostC by adding below to /fstab.

Pc 1: /NOVA-INST-DIR/instances nfs4 defaults 0 0

$mount - a - v$

$CheckthatNOVA/instances/directorycanbeseenatHostA$
ls -ld NOVA-INST-DIR/instances/
drwxr-xr-x 2 nova nova 4096 2014-04-10 14:34 nova-install-dir/instances/

Perform the same check at Pc 2 and Pc 3 - check perrmissions for write operation in nova.

- Source Host A:192.168.6.64 running nova-api,nova-schedular,nova-network
- Source Host B:192.168.6.65 running nova-compute
- ping 192.168.6.64 from source pc 1 to destination pc 2
- Update libvirt configuration
- libvrt migrate –live ub-vm1 kvm+ssh://192.168.6.65/system
- Migration Process Start
- Check the log in /root/etc/nova/libvrt/log

6.2 After Succesful Migration

After configure migration parameter the virtual machine is running in destination host.

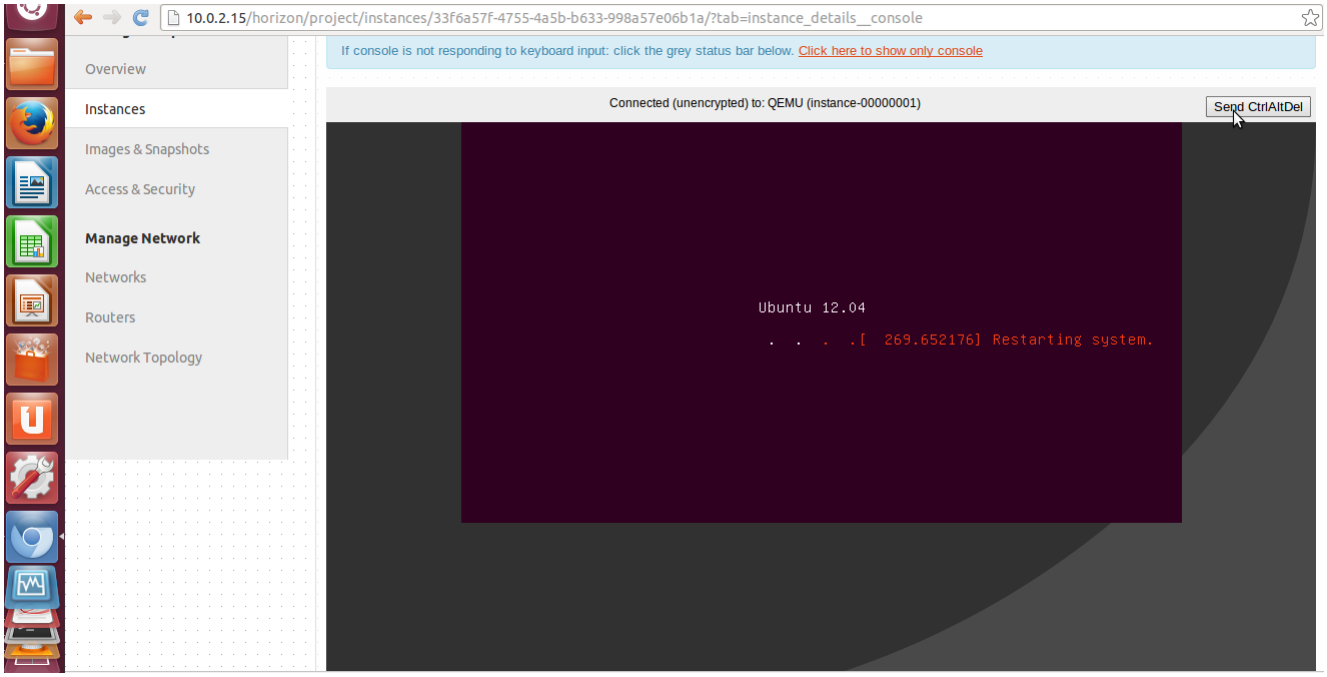

Figure 6.1: OpenStack Instances Succesful Running in web-browser in destination host

6.3 Result and Findings

The Performance of live migration using modified pre- copy algorithem is compared with traditional pre-copy based migration with the help of KVM. KVM is a built in migration tool for cloud computing environment. KVM Is a virtual machine monitor,also called as hypervisor.It allows several guest operating systems to be executed on the same computer hardware concurrently.

VM Name	Vm Size(Mb)	Flavors(M.B)	M. Time(s)
Mirantis	1800	512	160.62
Ubuntu 12.04	718	512	108.17
Open Solaris	605	512	100.32
Mini-OS	40	512	25.04

Table 1: Migration Time(S) of Different Vm in OpenStack Live Migration (Pre-copy)

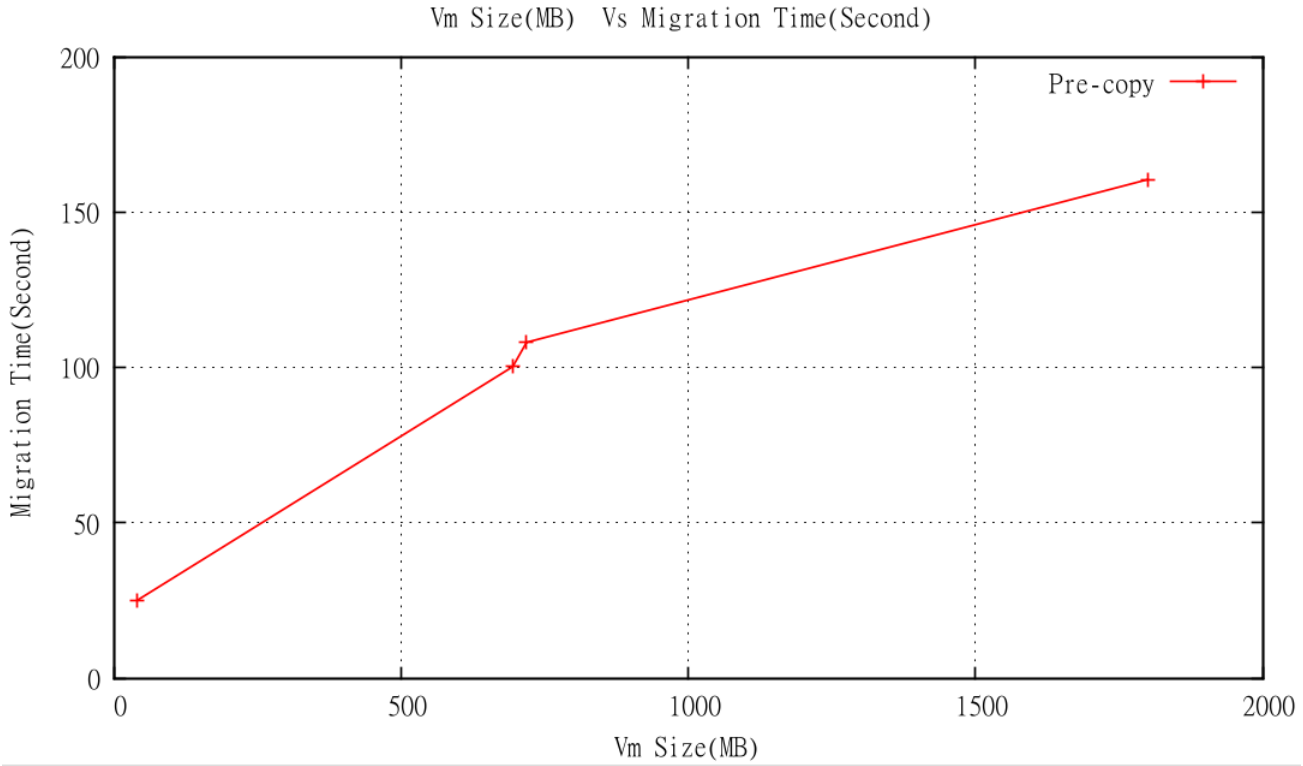

Figure 6.2: Migration Time(S) of Different Vm in OpenStack Live Migration (Pre-copy)

VM Name	Vm Size(Mb)	Flavors(M.B)	M. Time(s)
Mirantis	1800	512	118.75
Ubuntu 12.04	718	512	66.39
Open Solaris	605	512	64.35
Mini-OS	40	512	16.46

Table 2: Migration Time(S) of Different Vm in OpenStack Live Migration (Modified Pre-copy Algorithm

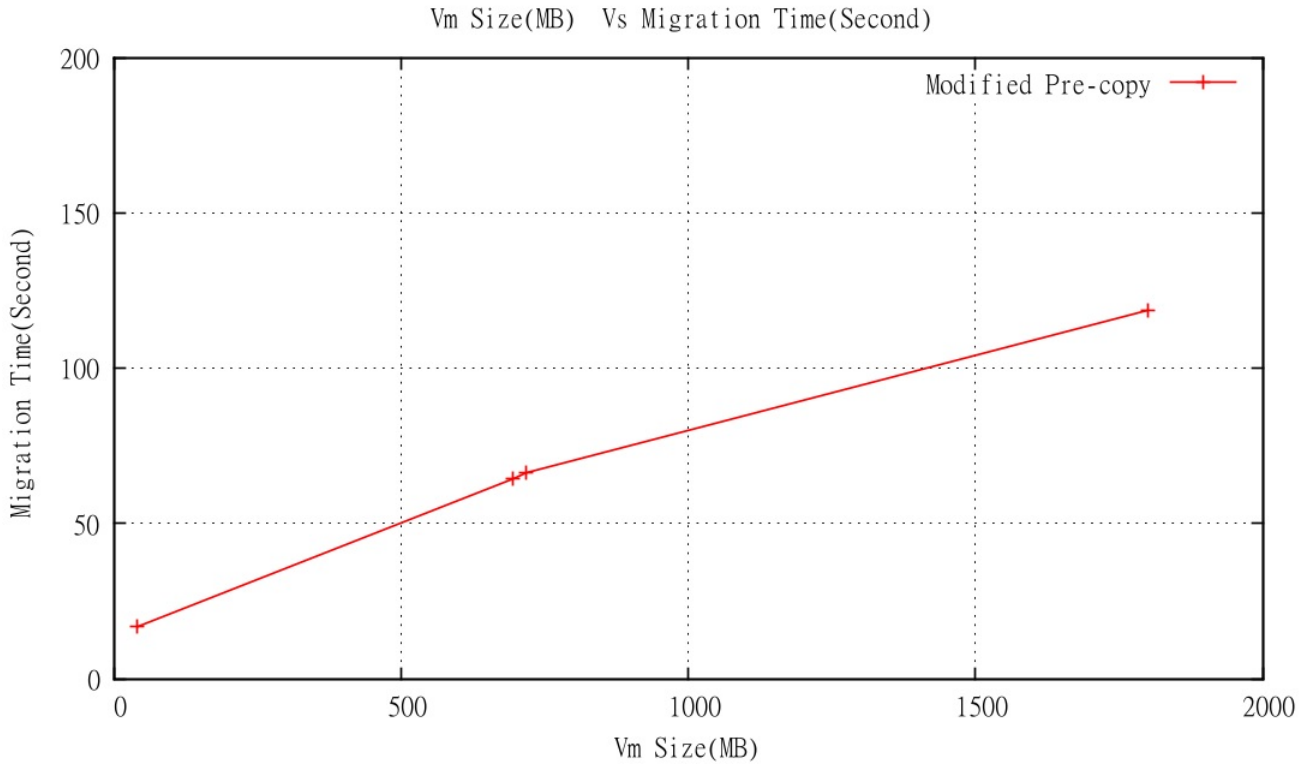

Figure 6.3: Migration Time(S) of Different Vm in OpenStack Live Migration (Modified Pre-copy Algorithm)

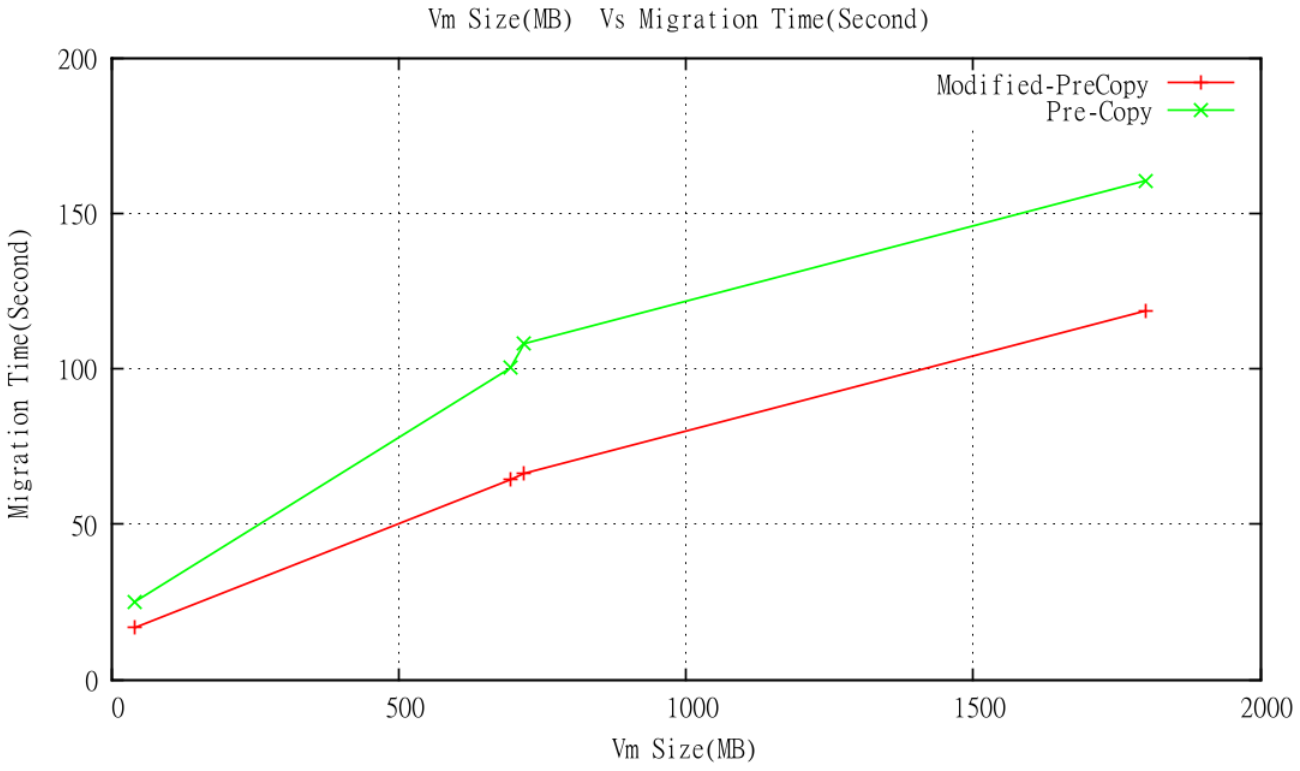

Figure 6.4: Migration Time(S) of Different Vm in OpenStack Live Migration (Modified Pre-Copy Vs.Pre-copy)

Chapter 7

Conclusion and Future Scope

In this work ,Live Vm Migration Algorithm is design by various migration cost parameters . Performance of modified pre-copy technique for live migration of virtual machines depend on mainly threshold value of pre-copy stages and memory dirty page rate.

Finally, Algorithem improves the performance of pre-copy approch by reduceing the migration time and the number of pages transferred in light workload environment with low dirty rate of memory pages.

Live migration of Virtual Machine face problem of performance overheads due to the unavailable service in the last stage of downtime .The performance overheads of live virtual machine migration is affected by processor resource,memory size and workload types.

Still, the resource reservation in destination node will not help to improve the migration efficiency in terms of total migration time, it is essential to avoid the live virtual machine migration failures because of unavailability of resources in the destination node. The sufficient resources in the source node can make more parallel number of live migrations and hence obtain better live migration efficiency.

Future scope will include design and implement efficient live vm migration mechanism to improve the live migration efficiency in the multiple virtual machine scenario.

Chapter 8

List of Paper Published

1. Title: Patel Pradip D.,Dr. M.B.Potdar,Dr.Madhuri D.Bhavsar,Mr. Miren Karamta Live Virtual Machine Migration Techniques In Cloud Computing: A Survey In International Journal of Computer Applications (0975 8887) U.S.A ,Paper Id:pxc3893453. Digital Library URL: http://ijcaonline.com/archives/volume86/number16/15070-3453
   Issue ISBN: 973-93-80879-96-7

2. Title:Patel Pradip D.,Dr. M.B.Potdar,Dr.Madhuri D.Bhavsar,Mr. Miren Karamta," Optimized Pre-Copy Live Virtual Machine Migration In Open Cloud Platform" In International Journal of Computer Science Communication (csjournals). (Submitted.)

Bibliography

[1] Divya Kapil ,Emmanual S. Pilli and Ramesh C. Joshi " Live VM Migration Techniques:Survey ". IEEE International Advance Cloud Computing Conference, 2013.

[2] Anja Strunk, "Costs of virtual Machine Live Migration". IEEE World Congress on Services Conference,2012.

[3] Kartik Gopalan and Michael R. Hines , "Post-copy based live virtual Machine Migration Using adaptive Pre-Paging and Dynamic Self -Ballooning," ACM , 2009.

[4] Steven Hofmen,Khaled Z. Ibrahim,Eric Roman,Costin Iancu,"Optimized Pre-Copy Live Migration for Memory Intensive Application"

[5] Umesh Deshpande,Michael R. Hines and Kartik Gopalan, "Post-Copy Live Migration Of Virtual Machines",.ACM SIGPLAN International Conference on Virtual Execution Environments March 2009.

[6] Diego Perez-Botero. "A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective".

[7] Dr. J Sharmila, P. Getzi Jeba Leelipushpam ,"Live Vm Migration techniques in cloud environment-A Survey",Proceedings of 2013 IEEE Conference on Information and Communication Technologies,2013.

[8] Dheeraj Rane , Shyam Patidar, "A Survey Paper on Cloud Computing",Second International Conference on Advanced Computing and Communication Technologies,2012.

[9] Dutta Borah Malaya ,Deka Ganesh Chandra, "A Study on Cloud Os", International Conference on Advanced Computing and Communication Technologies,2012.

[10] Masayuki Nakae ,Yuki Ashino,"Virtual Machine Migration Method between Different Hypervisor Implementions and its Evaluation, 26th International Conference on Advanced Information Networking and Applications Workshops,IEEE,2012

[11] Li Deng,Hai Jin,Song Wu, Xuanhua Shi, Xiaodong Pan,"Live Virtual Machine Migration with Adaptive Memory Compression" IEEE. 2009.

[12] Ming Zhao,Yangyang Wu, "Performance Modeling of Virtual Machine Live Migration", IEEE 4th International Conference on Cloud Computing,2011

[13] C. Ramesh,P. S. Constantine, P. Ben,S. L. Monica, C. Jim,and R. Mendel, Optimizing the migration of virtual computers, in 5th Symposium on Operating Systems Design and Implementation, SIGOPS Oper. Syst. Rev., vol. 36, Issue SI, pp.377-390, 2002.

[14] C. Christopher, F. Keir, H. Steven, H. Jacob Gorm, J. Eric,L. Christian, P. Ian, and W. Andrew, Live migration of virtual machines, 2nd conference on Symposium on Networked Systems Design Implementation - Volume 2: USENIX Asso- ciation, 2005.

[15] N. Michael, L. Beng-Hong, and H. Greg, Fast transparent migration for virtual machines, Annual conference on USENIX Annual Technical Conference Anaheim, CA: USENIX Association, 2005.

[16] H. Wei, G. Qi, L. Jiuxing, and D. K. Panda, High performance virtual machine migration with RDMA over modern interconnects, in IEEE International Conference on Cluster Computing, 2007, pp. 11-20

[17] L. Yingwei, Z. Binbin, W. Xiaolin, W. Zhenlin, S. Yifeng,and C. Haogang, Live and incremental whole-system migration of virtual machines using block-bitmap, in IEEE International Conference on Cluster Computing, 2008, pp. 99-106.

[18] B. Robert, K. Evangelos, F. Anja, S. Harald, and berg, Live wide-area migration of virtual machines including local persistent state,3rd International Conference on Virtual execution environment, San Diego, California, USA: ACM, 2007.

[19] http://docs.openstack.org/

[20] http://www.sebastien-han.fr/blog/2012/07/12/openstack-block-migration/

[21] http://www.python.org/

[22] http://docs.openstack.org/grizzly/basic-install/.