Implementing Private Cloud at IIT Roorkee: An Initial Experience

Susmita Horrow
Student
IIT Roorkee, Roorkee
India
hsusmita4@gmail.com

Sanchika Gupta
Student
IIT Roorkee, Roorkee
India
sanchigr8@gmail.com

Anjali Sardana
Assistant Proffessor
IIT Roorkee, Roorkee
India
dr.anjalisardana@gmail.cm

ABSTRACT

Cloud Computing has emerged as a promising technology to provide computing resources as public utility. Its features like cost effectiveness, pay per use and scalability have attracted many organization to adopt cloud computing environment in order to cut down IT cost. Hence market is blooming with the cloud vendors like Amazon, IBM who provide Infrastructure as a service. It saves the organizations form the complications of building infrastructure and hence the organization can concentrate on the quality of services and products to be delivered. Though the features like cost effectiveness and scalability have made cloud popular, at the same time, the features like multi-tenancy has refrained some of the organizations dealing with confidential information, from adopting cloud. This is because in the changed computing paradigm, the information of the client is present in the vendor's premises. This implies less control over the data. For educational institutes, which need to secure the confidentiality and privacy of data, a private cloud can be a good solution. Hence we propose private cloud architecture for educational institute which can provide Infrastructure as a Service to the students and teachers without compromising the confidentiality of information. The proposed architecture has been deployed using open-source cloud computing tool kit called OpenNebula. It has been compared with traditional computing infrastructure. Results show that our architecture increases resource utilization up to 85%, thus increasing the efficiency.

Categories and Subject Descriptors

H.3.4 [**Information Systems**]: Systems and Software- *Distributed Systems, Performance evaluation (efficiency and effectiveness)*

General Terms

Performance, Design, Experimentation, Verification.

Keywords

Cloud Computing, Mobility, Virtualization, Private Cloud

1. INTRODUCTION

Cloud Computing is defined by a large scale distributed computing paradigm that is driven by economies of scale, in which pool of abstracted, virtualized, dynamic-scalable computing power, storage, platform and services are delivered on demand to external customers over the Internet [1]. Cloud aims to deliver computing power, storage and platform as a public utility on pay per use basis. The features like scalability, flexibility, cost-effectiveness and ease of up-gradation have attracted many organizations to move to cloud. This section describes how cloud computing can be the need of cloud computing in the context of an educational institute followed by a brief discussion of the private cloud architecture.

If we closely study the usage of computing resources in an educational institute, we find that the demand of computing resource is non-uniform in different point of time. The traditional infrastructure does not scale well to meet the dynamic requirement of computing resources. At certain period of time such as project work, experiments and analysis period, report submission time, the number of active students is the highest. At this time, the traditional system becomes inadequate to meet those demands. In rest of the time, the computing resources are not fully utilized. If the infrastructure is built considering the maximum demand, then the computing resources are likely to be underutilized most of the time.

In present scenario, a student has to work on different kinds of computing platform which consist of various combinations of operating system and software tools. Hence valuable time of a student is devoted in building the platform. So there should be a system which can ease the process of establishing the computing environment.

Again in the traditional environment, upgrading any of the resources requires things to be started from the scratch.

Adopting cloud architecture can help to overcome the problems in the traditional computing environment discussed above. Cloud computing provides on demand and flexible provisioning of computing resources. Hence a student can avail the computing resource at any time which can be scaled down or scaled up as per the requirement. As cloud operates on the principle of virtualization of physical resources, it can support more number of users with effective utilization of computing resources.

However, moving to cloud comes with the risk of security. The user does not have the same level of control as that of standalone system in case of cloud, because user data present in the cloud provider's system. In an educational institute, there are

information related to research work, patents and other confidential issues. The confidentiality and privacy of data cannot be ensured completely if they are moved to public cloud.

Hence there is a need of a system which can provide cloud service to the organization or institute internally as which can be trusted for the confidentiality of data. This kind of systems can be realized by building private cloud.

A private cloud is solely owned by a single organization and managed internally or a trustworthy third party. As private cloud serves a single organization, enforcement and management of security policy become easier. Moreover, building private cloud is more cost-effective, because in a single organization, the requirements of the users are limited and properly specified.

In this paper, we propose private cloud architecture to provide Infrastructure as a Service to the students, teachers and staff and a mechanism to monitor and control the system. Providing remote access facility to the system will give mobility to the administrator of the cloud.

The rest of the paper is organized as follows. Section 2 describes architecture of the system. Section 3 describes the implementation details. Section 4 gives the details of the work flow of the system and describes VM cycle. Section 5 describes the related work. Section 6 concludes the paper.

2. ARCHITECTURE OF THE PROPOSED SYSTEM

Figure 1: Components of Cloud Infrastructure

Figure 1 shows the various components of cloud infrastructure and the interaction among them. Cloud infrastructure consists of a number of computer systems which are capable of provisioning a virtual machine whenever a request for computing resources arises. Cloud infrastructure works by interaction of three components, such as Front End, Cluster Node and Image repository.

The front end acts as an administrative node which handles the request for virtual machines sent by the users. It keeps the accounting information of the cluster nodes. Its job is to convert the user request form to a suitable template and then search for a suitable cluster node and instruct this node to launch the virtual machine according to the specification given in the template. The cluster nodes are set of computer systems with virtual machine manager installed in them. The job of the cluster node is to run the virtual machines. The image repository stores the images of the operating systems which are required in the virtual machines.

All the three components are connected through network and file sharing is enabled.

Figure 2 shows the series of events that take place starting from the user authentication and provisioning of virtual machine to the user.

1. User authenticates to the system
2. Receive client request
3. Perform validation of user request
4. If (validation successful)
   4.1 Search for suitable cluster node
   4.2 Create template
   4.3 Activate cluster node
   4.4 Deploy VM
   4.5 Return IP to Client
   Else
   Generate error.

Figure 2: Working Principle of Cloud Infrastructure

3. IMPLEMENTATION ISSUES

The proposed architecture is deployed using the open source cloud computing toolkit called as OpenNebula. OpenNebula provides the management console to carry out various functions of cloud infrastructure such as managing user requests, monitoring cluster nodes. As discussed in previous section, cloud infrastructure works by interaction of components for example the front end interacts with cluster node for invocation of virtualization software, front end interacts with image repository for transfer of operating system images to cluster nodes.

In order to carry out the interactions, OpenNebula provides a set of pluggable modules to interact with virtualization hypervisor, file transfer mechanisms or information services etc help the interaction among the components of cloud. These modules are called Middleware Access Drivers (MAD).Table 1 lists the modules along with their functions.

Table 1.Drivers provided by OpenNebula

IM_MAD	Information Manager Driver	Manages information about virtualization technology used.
VMM_MAD	Virtualization Manager Driver	Manages virtualization operations.
TM_MAD	Transfer Manager Driver	Manages transfer operation(ssh, shared, lvm)
IMAGE_MAD	Image Manager Driver	Manages image repository
HM_MAD	Hook Manager Driver	To execute the hooks
AUTH_MAD	Auth Manager Driver	To authenticate and authorize OpenNebula users

3.1 Virtualization

As the cluster nodes have to host the virtual machines, they need any kind of virtualization technology enabled in their systems. At present KVM, Xen, VMWare and Virtual Box are some of the popular virtualization technologies. OpenNebula provides drivers for each of the above virtualization technology. Choosing appropriate virtualization technology is very vital task. Each one of them needs different hardware requirements and each one of them has different features. One most important feature is live migration which allows moving the virtual machines from one cluster node to other without shutting it down. The requirement and the features of these technologies are listed in table 1.So appropriate technology is chosen according to the resources available and requirement of the system. Once the virtualization technology is finalized and installed in the cluster node, the necessary changes must be made in the configuration file in the front end. The fields IM_MAD and VM_MAD are filled according to the virtualization technology used. The front end reads these information and load appropriate virtualization manager drivers during start up.

Similarly while adding a cluster node as a potential host to launch virtual machine, the following arguments must be specified such as < IM_MAD, VMM_MAD, TM_MAD>.

The following table shows the system requirements and the features of various virtualization technologies.

Table 2.System Requirement and features of Virtualization Technology

Technology	Hardware Assisted Virtualization	Live Migration support
KVM	Required	Yes
Xen	Required	Yes
VMWare	Not Required	Yes
Virtual Box	Not Required	No

3.2 Networking

As cloud service is provisioned over the internet, configuration of network is an important aspect. As user will interact with the virtual machine from his/ her local machine, the virtual machine must be connected to the network. For this a virtual network has to be created. A single virtual network acts as a physical switch containing DHCP server. Virtual machines connected to particular virtual network are isolated from other network just like in case of physical. Creation of virtual network requires the following parameters <Name, Bridge, Type> where Name: Name of the Virtual Network

Bridge: Name of the physical bridge in the physical host where the VM should connect its network interface

Type: Ranged/Fixed. In case of Ranged type, the user has to mention the network address along with the subnet mask. In case of fixed type, a set of IPs are needed to be specified which are called as leases.

When a new Virtual Machine is launched, its virtual network interface (defined in the NIC section of the VM template) is connected to the pre-existing bridge or physical device specified in the Virtual Network definition. The next task is to assign IP and MAC address of the virtual machine to be able to connect to the network. The VM is assigned IP as per the specification provided in virtual network template. The MAC address assigned to the virtual machine is a transformation of the IP address.
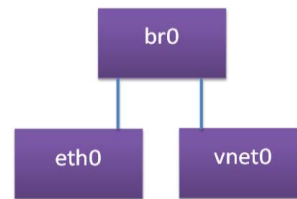
Figure 3: Networking inside the Virtual Machines

br0: bridge created by administrator during the installation process of OpenNebula (which may or may not have an IP)
eth0: real physical device (which doesn't have an IP)
vnet0: the virtual network device of the virtual machine, created by the hypervisor.
Here the bridge acts as a hub. All the packets that arrive through the physical device are forwarded over all the virtual network interfaces of all the running VMs. If a VM has configured with an IP of the network same as the network of the recipient, then that VM will capture the packet, otherwise it will simply ignore the packet.
In order to enforce security at network level, there are some drivers provided by OpenNebula. In order to enable these drivers, proper arguments must be specified while adding the cluster nodes. The arguments are given in table 2.

Table 3.System Requirement and features of Virtualization Technology

dummy	No network operation
fw	Apply firewall rules, ignore network isolation
802.1Q	Enables VLAN tagging
ebtables	Enables Ebtables rules
ovswitch	Restrict network access with Open Virtual Switch
vmware	For VMWare Networking infrastructure to provide 802.1Q compatible network

3.3 Remote Access

As specified before, the front end is in charge of invoking the cluster node and instructing it to host a virtual machine. Hence front end needs to have complete control of the cluster nodes. In order to have the remote access to the clusters, secure shell technique is used. SSH works on public key cryptography. As front end requires frequent interaction with the cluster node, it is desirable that, front end should not be prompted to enter password while connecting to the cluster nodes via SSH. This can be done by copying the public key generated by the front end in the cluster node. That key is added to the list of known users in the cluster node. So the cluster node will not prompt the front end to enter the password while initiating an SSH session. Hence the front end can perform all the necessary action in the cluster node like invoking the virtualization driver, copying image files etc without any interruption.

3.4 File Sharing

All the installation files are present in the front end. These installations include tools to control virtual machines and hosts and database. The cluster node needs to pull the required files from the front end during the configuration of virtual machines. Hence any file sharing techniques must be enabled. For this work, NFS (Network File Sharing) is used.

3.5 User Interface

In order to use the service provides by the cloud infrastructure, the user needs to register to the system. Once an account is created, the user is given access to the services. The proposed system provides different user profiles based on the different level of credibility. The user profile specification is given as follows:

$$U_j[i] = <N, T, V>$$

Where $U_j[i]$ represents i^{th} attribute for the user j

User type specifies the level of credibility. The credibility depends on the level of security.

User specification is defined as a vector of five parameters such as Operating system, Memory, Hard Disk, Processor, Software.

$$V_j[i] = <O, M, H, P, S>$$

Where $V_j[i]$ represents i^{th} specification for the user j.

<O, M, H, P, S> represent operating system, memory, hardware, processor and software respectively. Among these five parameters the first four are mandatory to specify. The last parameter is optional.

4. WORKFLOW AND VM CYCLE

4.1 Workflow of the system

Figure 4 demonstrates the workflow of the system. Once a client submits the specification, the following events take place.

1. Register image: In order to boot VM, an image of operating system is required. The images are stored in the image repository. A new image can be registered for use or preexisting image is can be used.
2. Register virtual network: To be able to communicate the virtual machine, its network must be configured. This is done by creating a virtual network. This is explained later in this paper.

3. Create template: Next a virtual machine template is created.
4. Instantiate.
5. The scheduler is invoked to search a suitable cluster node.
6. The disk image present in the image repository of the front end is copied to the cluster node as disk.0.
7. A disk image called disk.1 is created from isofile.
8. This disk.1 is copied unto the cluster node.
9. Deploymet.0 disk is made out of disk.0 and disk.1
10. The hypervisor is invoked by VM Mad and it is deployed.
11. Network configuration is done inside the virtual machine according to the virtual network template. Now the client can access the VM through vnc/ssh.
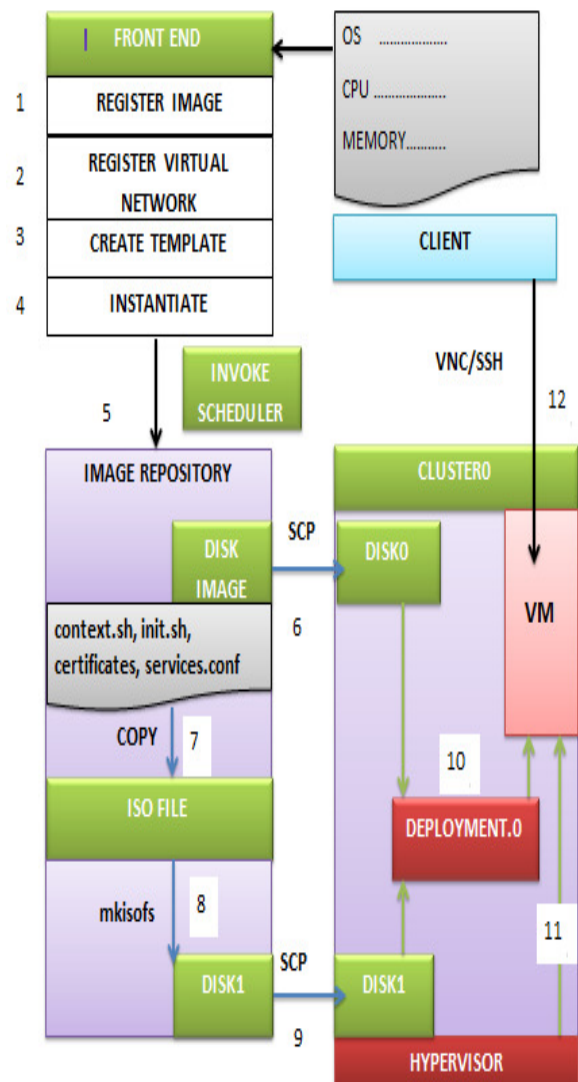

Figure 4: Workflow of the system

4.2 VM Cycle

Figure 5 explains the life cycle of a virtual machine. As soon as a template is submitted, the virtual machine is in **PENDING** state. During this state, the host manager monitors the cluster nodes. The scheduler searches for the suitable node to deploy the VM. When the cluster node is selected, then the VM comes to the **PROLOG** state. Then the deployment of the VM starts. In this stage, the Transfer Manager Driver copies the necessary disk images to the selected host. The hypervisor is invoked using the Virtualization Manager Driver. When it is done, the VM is in **BOOT** state. During this state, network configuration takes place inside the VM. Then the VM becomes to the **RUNNING** state. When the client issues shut down signal, the Virtual Machine Driver will send the shutdown command to the underlying virtual infrastructure. The VM is now in **SHUTDOWN** state. In **EPILOG** state the Transfer Manager Driver is called again. It copies back the images that have *SAVE=yes* option and deletes images that were cloned or generated by **MKSWAP**.
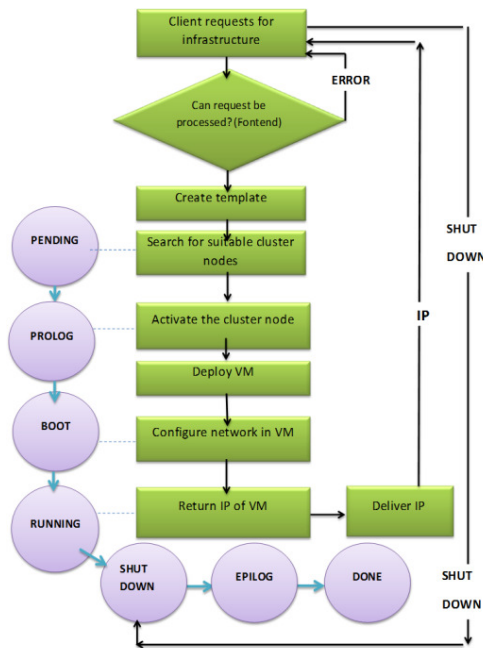
Figure 5: VM Cycle

5. EVALUATION

To perform experiments, a private cloud was deployed in the computer laboratory with 20 computers. One of the computers was made front end and other computers were treated as clusters. The aim was to build a private cloud to provide mobile IaaS with existing hardware. Table 3 gives the hardware and software specification.

Table 4.Hardware and Software Specification

Operating System	Ubuntu 11.10
File Sharing	NFS
Hypervisor techniques used	Virtual Box
RAM	2 GB
Hard Disk	300 GB
Processor	Core2 Duo

As the workload on the system of the lab is not constant throughout the day, the working hours of the lab has been divided into four slots of three hours such as 8am-11am, 11am-2pm, 2pm-5pm and 5pm-8pm. We have evaluated the system performance by measuring CPU utilization in two scenarios, during heavy workload and during normal workload.

A. *Heavy workload*

Figure 6 shows the comparison of cloud based and traditional system in terms CPU utilization. It is quite evident that Cloud based systems are system utilization is improved about 85%.
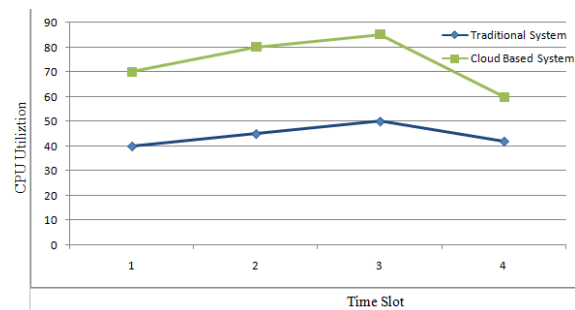

Figure 6. Graph showing comparison of traditional and cloud based system in terms of CPU utilization in heavy workload
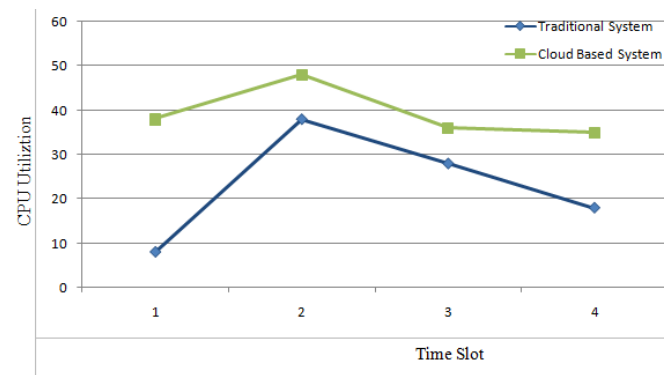
B. *Normal workload*

Figure 7. Graph showing comparison of traditional and cloud based system in terms CPU utilization in normal workload

Figure 7 shows the improved CPU utilization in cloud based system as compared to the traditional system. Because of the flexibility provided in our system, the users could work at the places of convenience. Therefore we observe better resource utilization in our system.

6. RELATED WORK

North Carolina State University has done significant work in deploying private cloud for educational purpose. The University has developed a lab called virtual computing lab which provides Infrastructure as a service to the students and teachers. The architecture is more sophisticated and is deployed keeping many aspects of computing services.

But our work attempted to build cloud using the resources available in public domains to prevent resource underutilization.

7. CONCLUSIONS

In this work we have deployed and evaluated the private cloud architecture for educational purpose. We conducted experiments and evaluated the performance in terms of CPU Utilization. We compared the system with the traditional computing infrastructure. Results show that our proposed system shows better CPU utilization as compared to the traditional system. The future work is directed to deploy the prototype in large scale.

8. ACKNOWLEDGMENTS

We are grateful to all the persons involved in mailing lists, development and documentation of OpenNebula project.

9. REFERENCES

[1] Foster,I, Zhao,Y., Raicu, I, and Lu,S. 2008. Cloud Computing and Grid Computing 360-Degree Compared. *Grid Computing Environment Workshop* (Austin, TX, Nov 12-16, 2008). GCE'08.1-10. DOI=http://*dx.doi.org/10.1109/GCE.2008.4738445.*

[2] Averitt, S., Bugaev, M., Peeler, A, Shaffer, H., Sills, E, Stein, S., Thompso, J., and Vouk, M. 2007. Virtulal Computing Laboratory (VCL).In *Proceedings of the International Conference on Virtual Computing Initiative* (IBM Corp., Research Triangle Park,NC, May 7-8,2007).1-16.

[3] Buyya,R., Yeo, C.S., and Venugopal,S. 2008. Market Oriented Cloud Computing: Vision, Hype and Reality for Delivering It services as Computing Utilities. *In Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications* (Dalian, Sept 25-27, 2008).5-13. DOI= http://dx.doi.org/10.1109/HPCC.2008.172.

[4] Moothoor,J, and Bhatt,V. 2009. A Cloud Computing Solution for Universities: Virtual Computing Lab, Case study of North Caroline State University's Virtual Computing Lab.

[5] Montero,R.S. 2010. Deployment of Private and Hybrid Clouds Using OpenNebula/ RESERVOIR. *Open Grid Forum 28*, (Munich, Germany, March 15-18, 2010).

[6] Montero,R.S. 2010.Building Clouds with OpenNebula 1.4. *CESGA Santiago de Compostela, Spain, January 2010.*

[7] Velte, A.T., Velte, T.J., and Elsenpeter,R. 2011. *CloudComputing – A Practical Approach.* Wiley Publishing,Inc.

[8] Sosinsky, B. 2010.*Cloud Computing Bible.* McGraw-Hill Companies.