Architectural Comparison and Implementation of Cloud Tools and Technologies

N. Nagar and U. Suman

Abstract—**Service oriented architecture is a collection of services that communicate with each other to provide flexibility in system development and deployment. Cloud computing emphasizes on various service oriented architectures with the help of tools, which reduces the infrastructure cost of users. There are varying range of cloud tools and technology such as Eucalyptus, OpenNebula, Nimbus and OpenStack etc. The comparative study of different cloud tools and technology in this paper is performed on the basis of certain cloud parameters for the deployment of clouds. The concepts, architecture and implementation of cloud tools are also discussed here. The comparative study will be helpful to select an appropriate deployment strategy for users.**

Index Terms—**Cloud computing, eucalyptus, nimbus, openstack, opennebula.**

I. Introduction

IT practitioner always requires rapid application development and deployment but it is very difficult in today's highly competitive and rapidly changing global business environment. The computing of information society has been revolutionized from distributed to cloud computing. The recent trend of cloud computing is proving a bonus to IT users and developers. It speeds up software deployment strategies by reducing time and effort required to prepare the application. Cloud computing is essentially a form of distributed computing that allows users' ability to tap into a vast network of computing resources through the Internet. Cloud computing refers the means, where everything comes from computing power to application, infrastructure and the business processes. It is delivered as service whenever and wherever it is required.

Cloud computing is the integration of different services such as, Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and so on. SaaS is based on the concept of renting an entire finished application from a service provider rather than buying, installing and running that software and also licensing of software is not a critical issue for the users. PaaS is all about providing a platform to receive the computing power, storage and networking infrastructure as a service via public Internet upon which the applications can be developed and executed. IaaS offered computing and storage capability on demand. The difference between PaaS and IaaS is that IaaS vendors deploy virtualization technologies to provide the compute power, whereas PaaS allows accessing an environment upon which applications can be deployed. In IaaS, a virtual machine is created for the user application and all other things required by the application rather than being restricted to a certain development environment. The user can select own choice of OS images, development environment and host it on the IaaS vendor infrastructure.

There are different tools available for the needs of individual or the organization for deployment of their cloud infrastructure plan. Cloud infrastructure plan may include tools such as, Elastic Utility Computer Architecture for Linking Your Program to Useful System (Eucalyptus), Open Nebula, Open Stack, Nimbus, ABICLOUD, Cloud Sim, Cloud Stack etc. These tools provide different services (i.e., SaaS, laaS and PaaS) according to the need of users. Cloud computing deployment depends upon whether the cloud is private, community, public or hybrid [1]. Private cloud is operated for a particular organization, whereas public clouds are available to the common public or large groups of Industries. Community clouds are mutual by a number of organizations while hybrid clouds combine public and private elements in the same data center.

Tools such as Google's App engine, Microsoft Live Mesh, Sun network.com (Sun Grid) are based on service-based infrastructure. These tools are different in storage capability, compute service, Web APIs etc. The infrastructure of these tools is based on the type of services provided by these tools such as Google's App engine based on PaaS, Microsoft Live Mesh based on IaaS and Sun Grid based on IaaS [2]. These tools focus on only private cloud computing environment.

Open Nebula, Eucalyptus, Nimbus and ABICLOUD tools are compared with different parameters such as cloud type, web interface, deployment manner etc. in 2009 [3]. Most of the important parameter of comparison is not included such as binary package support, scheduling and policy algorithm used in these tools, suggested and minimum configuration required by tools. These parameters are imported to manage cloud in any infrastructure.

To overcome above problem Open Stack with three other tools such as Eucalyptus, Open Nebula, and Nimbus tools will be analyzed and compared with latest and advanced features. There are different comparison criteria such as cloud types, compatibility, deployment strategies, scalability, hypervisor support, security management, scheduling and policy algorithm, web interface, supported type, supported language, reliability, SSH key management, disk allocation, service type, latest release name etc. The architecture and deployment strategies for better understanding of these tools are also discussed in this paper.

The introduction, architecture and implementation of these tools are discussed in Section II. The Section III

Manuscript received May 4, 2013; revised July 23, 2013.

N. Nagar is with the International Institute of Professional Studies, Devi Ahilya University Indore, India (e-mail: nitin28nagar@gmail.com).

describes the comparison of underlying tools with different parameter such as SSH management, cloud type, OS support, language support etc. The applications, pros and cons and affecting issues of above tools are presented in Section IV. Concluding remarks and future work are stated in Section V.

II. Cloud Tools and Technologies

Cloud tools and technologies can be used to solve business solution depends on the organizational needs. The implementation of cloud tools is an important aspect. Each tool has different strategies of deployment. For example, the deployment can be performed through binary packages such as Cent OS, Open SUSE, Debian, Fedora and deployed with Ubuntu Enterprise Cloud (UEC). Ubuntu 9.04 (Jaunty Jack lope) and any higher version of Ubuntu or UEC are required to deploy Eucalyptus. Ubuntu 10.04 or Cent OS 5.5 is highly recommended to install Open Nebula. Nimbus and Open Stack installed with any preferable Linux or Ubuntu version. Cloud tools and technologies such Eucalyptus, Open Nebula, Nimbus and Open Stack are discussed in the following subsection with their architecture, implementation and features.

A. Eucalyptus

Eucalyptus was originated from the University of California at Santa Barbara, which is now supported by eucalyptus incorporation. It provides an integrated set of APIs that are compatible with Amazon Web Services (AWS), Amazon's Elastic Cloud Computing (EC2), Amazon's Simple Storage Services (S3) and EBS (Elastic Block Storage). The architecture and its implementation are discussed in the following subsections.

1) Architecture: There are various components of Eucalyptus such as, CLC (Cloud Controller), Cluster Controller (CC), Data Center Manager (DCM), Node Controller (NC) and Walrus. CLC provides the web user interface (an http server on port 8445) and implement on Amazon EC2 APIs. There should be only one CLC in installation of UEC. CC is used to manage the collection of resource nodes. DCM provide by Ubuntu Eucalyptus – CC package. NC runs on nodes, which host VM. Walrus component is used to implement S3 APIs and is useful to store VM Images and user storage using S3 bucket put/get abstraction. EC2 provides a virtual computing environment that enables a user to run Linux-based applications. The working of Eucalyptus is shown in Fig. 1, in which the user can either directly interacts with cloud controller through web services, SOAP and CLI or user can demand VM through cluster controller. CC control cluster A and cluster B, even CC can control multiple clusters. Each cluster contains multiple node controllers, VM and hypervisors to host the VM images. Now user can either create a new Amazon Machine Image (AMI) containing the applications, libraries, data and associated configuration settings, or select from a library of globally available AMIs and then upload the created and selected AMI to S3 [2].

2) Implementation: A basic UEC pre-requisition must be installed before deployment of Eucalyptus tool. UEC install *euca2tools* to manage the cloud. Deployment spans into three servers; two servers (server1 and server2) will run a 64-bit server version and the third server will run a desktop 64-bit version (client1). On server1, CLC, SC, Walrus and CC will be installed and on server2, hypervisors and node controllers will be installed. The server1 and server2 must be VT enabled. User can setup Eucalyptus on a single server (i.e. CC and NC on single machine). The configuration of Eucalyptus in the deployment of VM, user has to use *euca2tools* front end to request a VM by following commands:

```
user@user1:~$ mkdir ~/.euca
user@user1:~$ cd ~/.euca
user@user1:~$ unzip –key-name-zip.zip
user@user1:~$ chmod 0700 ~/.euca
user@user1:~$ chmod 0600 ~/.euca/*
```

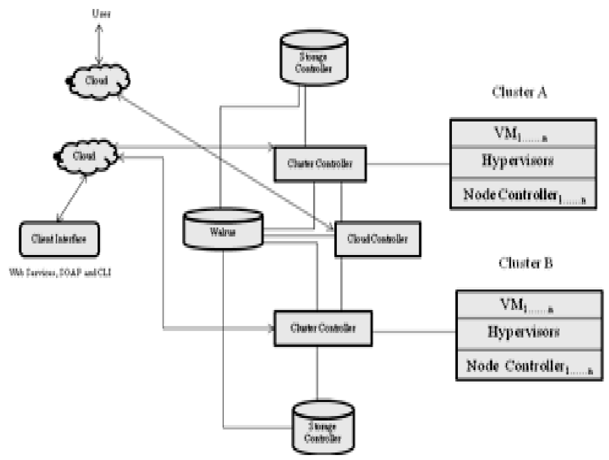

Fig. 1. Structure of eucalyptus

The compute node sets up network bridging to provide *NIC* with a virtual *MAC* [6]. To create *keypairs,* followings commands can be used.

```
user@user1:~$euca-add-keypair mykey  | tee mykey.private
user@user1:~$chmod 600 mykey.private
```

The instances that are accessible with newly generated private keys can be run as:

```
user@user1:~$ euca-run-instances -k mykey –n < number
of instances to start > <emi-id>
user@user1:~$ euca-describe-instances
```

On the head node, *DHCP* is set up with *MAC/IP* pair. *VM* is spawned on the *VMM*. For providing authorizing security groups and allocating *IPs*, The following command can be used:

```
user@user1:~$ euca-authorize -P tcp -p 22 -s 0.0.0.0/0
default
user@user1:~$ euca-allocate-address
user@user1:~$ euca-associate-address <IPfromallocate>-i
<Instance ID>
user@user1:~$ssh-i mykey.private root@ <accessible-
instance-ip>
```

Now, user will perform *SSH* directly into the *VM* [10].

B. Open Nebula

Open Nebula is an open source toolkit for cloud computing, which manages a data center's virtual infrastructure to build public, private and hybrid laaS clouds. And it proposes virtual environment accessible through two different interfaces open cloud computing interface (OCCI) or EC2 and through web interface, sunstone. The OpenNebula research project started in back 2005 by Ignacio M. Llorente and Ruben S. Montero. First public released of software was in March 2008 [8].

1) Architecture

It provides a cloud controller component, which is able to manage different instances of Open Nebula. It includes the ability to manage zones and virtual data centers and to redirect clients to their designated zone. Other components of Open Nebula are interfaces, scheduler, Open Nebula core, Open Nebula APIs and driver APIs [5]. Claudia is the latest feature added with interface. Open Nebula, provide management persistence through advanced deployment tool Open Nebula Express (which makes deployment more easy). Driver APIs interact with different components such as compute, cloud, storage, cloud-security and virtualization. This component provides different functionalities according to the need of client. Client can use either XML-RPC or Open Nebula cloud APIs. Fig. 2 represents the working of Open Nebula cloud computing.

The key features and benefits of Open Nebula (In private cloud) in the latest release provides user management, VM management, virtual network management, service management, infrastructure management, information management, scheduling, user interface, operations center, infrastructure management etc. The user management functionalities are authentication of framework, it also provides multiple cloud user and administrator roles and secure multi-tenancy. VM management provides image repository with catalog and functionality for image management, access control and creation of images. Virtual network management provides virtual network management to interconnect virtual machines and used to sharing virtual networks. VMM used to support multiple hypervisors in the same physical box. It also provides automatic configuration of virtual machines.

The main functioning of service management is the auto-configuration at boot time and support for Microsoft windows and Linux machine images. Infrastructure management provides management of physical hosts and creation of logical clusters. Information management supports multiple hardware platforms such as fibrechannel, iSCSI, Network Attached Storage (NAS) and so on.

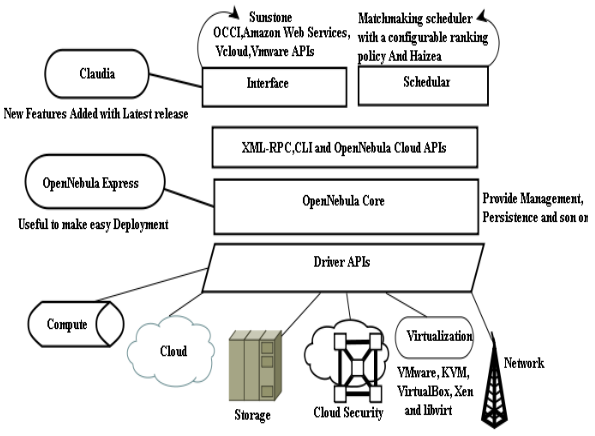
Sunstone
OCCI,Amazon Web Services, Vcloud,Vmware APIs
Matchmaking scheduler with a configurable ranking policy And Haizea
Claudia
Interface
Schedular
New Features Added with Latest release
XML-RPC,CLI and OpenNebula Cloud APIs
OpenNebula Express
OpenNebula Core
Provide Management, Persistence and son on
Useful to make easy Deployment
Driver APIs
Compute
Cloud
Storage
Cloud Security
Virtualization
VMware, KVM, VirtualBox, Xen and libvirt
Network

Fig. 2. Structure of nimbus

Open Nebula (in hybrid cloud computing) also provides additional features such as local infrastructure with remote cloud resources and also support new feature Cloudbursting. In public cloud computing that provides capabilities for exposing cloud interfaces to the private infrastructure. The features include such as cloud interface, now user can interact with cloud with different interfaces such as OGF OCCI and Amazon EC2. It also support for simultaneously exposing multiple cloud APIs.

In the latest release of Open Nebula, integration and production feature have been enhanced with some new features such as infrastructure abstraction, adaptability and customization, interoperability and standards, openness, programming interfaces, security, robustness, fault tolerance and performance. Open Nebula ecosystem also enhanced with new feature in latest release. The new components enhancing the functionality provided by the Open Nebula cloud toolkit or enabling its integration with other products such as, v Cloud API, Open Nebula express, libcloud, web management console, Haizea scheduler, and Deltacloud adaptor [8].

2) Implementation

For deployment of Open Nebula, *Unbuntu 10.04* or *Cent OS 5.5* can be used and deployment can be performed through one of the Open Nebula's component *Open Nebula Express*. In installation of Open Nebula, the installer must have capable of deploying *Ubuntu 10.04 – KVM – NFS, Ubuntu 10.04 – KVM – SSH, Cent OS 5.5 – Xen – NFS, Cent OS 5.5 – Xen – SSH and RHEL 5.5 – KVM – SSH.*

Physical cluster nodes are deployed on Open Nebula, Where one node work as front end and other nodes treated as worker nodes. Therefore at least two nodes for deployment are required. On the time of deployment of Open Nebula express, it is highly recommended that first, OS must be clean install. Second, worker node must have the network bridging before installation. Currently distributions are supported for *Ubuntu 10.04* and *CentOS 5.5* only [8]. VM in a configuration of OpenNebula: Users use *SSH* login to the head *node*.

User has to use *oneadmin* right with these commands.

```
user@frontend:~$ sudo -iu onadmin
oneadmin@frontend:~/ cd one-templates
oneadmin@frontend:~/one-templates
```

After enter in one-template, user can find out the list of directories with *ls* command. And here user can find the already available images list in one-template; here *tty-linux* image is used to create *VM*. User can use *onevm* command to request a *VM*.

```
oneadmin@frontend:~/ one-templates $ onevm create
ttylinux.one
oneadmin@frontend:~/ one-templates$ onevm list
```

The output will come in this particular format.

```
ID USER   NAME STAT CPU MEM HOSTNAME TIME
```

The *VM* templates disk image is copied and configuration within *SSH* directly on the host node.

```
oneadmin@frontend:~/ one-templates $ onehost create
frontend im_kvm vmm_kvm tm_ssh
oneadmin@frontend:~/one-templates$ onehost list

ID NAME CLUSTER RVM TCPU FCPU ACPU TMEM
FMEM STAT
```

The *oned* process on the head node uses *SSH* to log into a compute *node*. The compute node sets up network bridging to provide a virtual *NIC* with a virtual *MAC* [6]. Files needed by *VMM* on the compute node. And it will be pulled to the compute *node* via *NFS*.

```
oneadmin@frontend:~/one-templates$ onevm show 1
```

This command provides the information of virtual machine 1 information, virtual machine monitoring, virtual machine template, nic in detail.

```
oneadmin@frontend:~/one-templates$ onevm show ttylinux
| grep IP
IP=192.168.X.X
oneadmin@frontend:~/one-templates$ ssh
root@192.168.X.X
```

Now user can *SSH* directly into *VM*.

C. NIMBUS

Nimbus is the combination of open source tools, which provides IaaS cloud computing solutions to users. It allows users to lease remote resources by deploying VMs on those resources and configuring them to represent an environment preferred by the user. It is formally known as Virtual Workspace Service (VMS); generally workspace service is a technical component in the software collection. Users have three major options such as, it provides functionality to use single nodes in clouds as a client, user can launch auto-configuring clusters and user can build their own cloud using the workspace service.

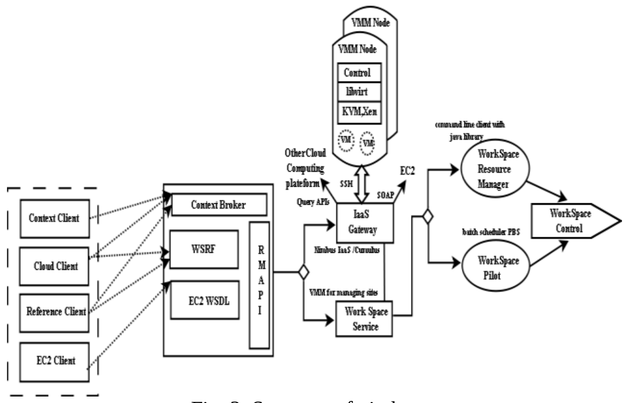

Fig. 3. Structure of nimbus.

1) Architecture

The architecture of Nimbus shown in Fig. 3 include different component such as, work service, web services resource framework (WSRF), cumulus, SOAP and query APIs, RM-API, cloud client, reference client, work space pilot, work space control, content broker and context agent and so on. Work service is independent VMM for managing sites. WSRF is web service wrapping tool that allows scientists to rapidly expose applications, scripts and workflows as web services. It is also useful for remote protocol implementation. Cumulus is open source implementation of the Amazon S3 REST APIs. SOAP and query APIs are implemented on the based on EC2. RM-API is bridge between remote control security and specific site manager implementation. Reference client can be used to expose the entire features set up in the WSRF protocol as command line client with java library. It also applied for advanced uses, scripting and portal integration etc. Workspace pilot is used to integrate VMs with resources. Nimbus includes batch scheduler known as PBS. Workspace control is used to implement VMM and network specific task on each hypervisors. Content broker allows users to repetitively and automatically coordinates large virtual cluster launches. Context agent lives on VMs and interacts with the context broker at VM boot [11].

The major features of Nimbus are storage cloud service, remote deployment and lifecycle management of VMs, compatibility with Amazons network protocols, supports X509 credentials certificate, easy to use cloud client. It also supports fast propagation, multiple protocol support, and flexible group management, per-client usage tracking, per user storage quota, flexible request authentication and authorization, easy user management. Configuration management, one-click clusters (contextualization), workspace client, VM network configuration (deployment request), local resource management plugins, Xen and KVM plugins, VM fine-grain resource usage enforcement (resource allocation) are also provided by Nimbus [11].

In latest release of Nimbus cloud computing tool include new features such as, propagation, VM image caching, libvirt template, import keypair is implemented in the EC2 protocols and so on.

2) Implementation

It is highly recommended when user first time installing Nimbus. User will have working cloud configurations that serve remote users using the cloud-client, EC2 clients such as boto, and S3 clients such as s3cmd. The different constructing steps of VM in configuration of Nimbus are *Service Dependencies, Service Installation, Service Installation, and Install DHCPd and configure networking etc.* The steps for constructing a VM in configuration of Nimbus are as follows.

Service Dependencies: All the dependencies must be installed before deployment of service node. Service node interacts with clients and manages backend hypervisor nodes. There are few system dependencies and installation requirements, which includes Sun Java 1.5 or later, python 2.5 or later (but not 3.x), Apache ant 1.6.2 or later and GCC.

Service Installation: After installation of service node, now user can install central services and image repository (*Cumulus*). It used to access cloud as a remote client. This will however only work in *fake mode*, which means that the service is only pretending to start *VMs*. The command to install service node are as follows.

```
user@user1:~$wgethttp://www.nimbusproject.org/downloa
d/nimbus-2.8-src.tar.gz
user@user1:~$ tarxfz nimbus-2.8-src.tar.gz
user@user1:~$ cd nimbus-2.8-src/
```

User can refer to installation directory called

$NIMBUS_HOME. To install, run this command from the Nimbus source directory.

TABLE I: COMPARISON OF CLOUD TOOLS AND TECHNOLOGIES

Tools / **Properties**	**Eucalyptus**	**OpenNebula**	**Nimbus**	**OpenStack**
Cloud types	Private	Public, Private and Hybrid	Private and Public	Public, Private and Hybrid
Storage compatibility	Elastic CC S3	Open Multi Platform	Amazon EC2 and other tools	EC2, OpenStack API
Deployment strategies	Command Line tool as euca2tool and Web UI	Web Interface	Command Line	Command Line tool euca2tool And NOVA API
Scalability	Scalable	Dynamic Scalable	Scalable	Massively Scalable
Hypervisor support	VMware, KVM, Xen and ESX, Virtio	VMware, KVM, VirtualBox, Xen and libvirt	Xen 3.x or KVM and bash, ebtables, libvirt	Xen, KVM, Hyper-V QEMU, UML (User Mode Linux), XenServer, LXC
SSH Management	SSH	SSH or NFS	SSH and for server side Java Secure Channel (JSch)	SSH
Scheduling and policy algorithm	Round Robin And GREEDY	Rank matchmaker, packing, striping, load-aware, affinity-aware	PBS and SGE	Diablo v3
OS Supported	Linux, can host Linux and Windows VMs	Linux	Linux	Linux
Supported languages	Java, C	C++, C, Ruby, Java, Shell script, lex, yacc	Java1.5 +, Python2.4 +	Python
Version Under	Proprietary, GPL v3	Apache License version 2	Apache License version 2	Apache License
Security Management	X509 certificate	In latest release it support X509 certificate	X509 certificate	X509 certificate
Image management	YES	YES	YES	YES
Linux VM Management	YES	YES	YES	YES
Binary packages Support	CentOS, openSUSE, Debian, Fedora and UEC	Ubuntu 10.04 and CentOS 5.5	Ubuntu 10.04	Ubuntu 11.04
Storage Management	Walrus	S3 (Simple Storage Structure),scp-wave	Cumulus or Distributed Storage	Amazon S3 and SWIFT
Web service and API	Web services	OGF OCCI API, EC2 query, vCloud	WSRF, EC2 WSDL interfaces	EC2 and OS APIs
Disk allocation	Eager Disk allocation	Support both Lazy and Eager Disk allocation	Do not know	Do not know
Service type	IaaS	IaaS	IaaS	IaaS
Latest release name	Eucalyptus *v 3.0*	OpenNebula *v* 3.0.0	RC2 Nimbus *v* 2.8	Cactus
Suggested configuration requirement	VT Enables Processor 64-bit, memory 2 GB RAM, disk space 200 GB.	A 64 core cluster will typically run around 80VMs, each VM will require an average of 10GB so disk space around 1TB required.	2.2 GHz AMD64 Processors for each nodes, 4 GB RAM, and 80 GB local disk,it depends upon the need.	Processor 64-bit x86, memory 12 GB RAM,disk space 30 GB and 1 GB NIC.
Minimum configuration requirement.	VT Enables Processor 32-bit x86, minimum memory 1 GB RAM, disk space 40 GB.	VT Enabled Processor, 32 bits x86, minimum memory 10GB of disk space. it depends upon the requirement.	16 Intel Xeon 2.40 GHz processors (VT enable) with up to 2.5 GB RAM. 10 GB of disk space.	Processor 32-bit x86, memory 2 GB RAM, disk space 5 GB.
Developer	Eucalyptus Systems, Inc	OpenNebula Community	Kate Keahey, Tim Freeman, et al	NASA and RackSpace
Reliability	Less Reliable	Less Reliable	Rollback host	More reliable as compare to others tools
Structure	Module	Module	Light weight Component	django-OpenStack
Unique Features	Emulate Amazon AWS	Support for multiple types of users	Suited for small and mid-sized enterprises	Compatible and Connected

user@user1:~$. / install $ NIMBUS_HOME

Now that the services are installed, you can try to start them for the first time.

user@user1:~$ cd $ NIMBUS_HOME

Now that Nimbus is installed and running, user can move on to testing with a real client. But first, user needs to create a user account to test it.

user@user1:~$./bin/nimbus-new-user-d/tmp/newuser user2@user2.com

Now, user can use service provided by Nimbus cloud client. Before that user first, need to download the credential from the nimbus site. After downloading credential, user needs to install the files produced earlier by *nimbus-new-user*. Cloud properties file needs to be placed into the extracted cloud client's conf/ directory. The *usercert.pem* and *userkey.pem* files need to be installed into *~/ nimbus/* in users home directory.

Install DHCPd and configure networking: Now user can install central *DHCPd* daemon (or integrate with an existing one) and configure the networking addresses that give to VM on boot time.

Install VMM Software: Now, user can install the Nimbus *VMM* software (*workspace-control*) and any of its dependencies (including Xen or KVM as well as libvirt, if these are not present already). Workspace-control contains scripts that will help to test the Xen/KVM/libvirt installation. It will work with the programmatic access patterns that Nimbus uses. Here the accounts privileged choose from the two mechanisms such as *Privileged account* and *Super user account. Privileged account*, it picks an account to make privileged (using the sudo program). It will refer to an account of the type name nimbus with terminal prompts such as *nimbus $. Superuser account*, the root account is necessary to install dependencies on the VMM nodes and it refer to an account of the type name root with terminal prompts such as *root #*.

SSH Setup: It configured *SSH* on cluster so that the Nimbus components can properly and securely communicate. It generate an *SSH keypair* on the service node with following commands.

```
user@user1:~$scp~/. ssh/id_rsa.pubnimbus@vmmode:
service_rsa.pub
user@user1:~$scp~/.ssh/id_rsa.pubnimbus@vmmode:
service_rsa.pub
user@user1:~$ ssh nimbus@vmmode
user@vmmode:~$ ssh-keygen
user@vmmode:~$ exit
user@service:~$ scpnimbus@vmm1:~/.ssh/id_rsa.pub./
vmm1_rsa.pub
user@service:~
$ cat ./vmm1_rsa.pub >> .ssh/authorized_keys
```

Final Tests: user can remotely test the system with the cloud client; a virtual machine will be started. Now, user can now performed *SSH* directly into *VM* [11].

D. Open Stack

Open Stack is an IaaS computing project jointly presented by NASA and the Rack Space [8]. The Rack Space start cloud formed in 2005 and it was rewritten in 2009. And again in 2010, they rewrite cloud servers and open source and released there first open source cloud computing tool Open Stack. NASA, don't just send stuff in space and also research institution for the government of US. They found the problem such as liability; scalability and so on in Eucalyptus and other tools. To overcome the underlying problems in tools, NASA decided to release its own object Nebula (not related with Open Nebula) in Feb 2010 [9].

1) Architecture

The architecture of Open Stack includes different component such as, compute (NOVA), object storage (SWIFT) and image service (Glance). NOVA is cloud computing fabric controller. It uses the event let (it is concurrent network library for python) and twisted framework (it support TCP, UDP, SSL/TLS, IP multicast). NOVA relies on the advanced message queuing protocol (AMQP). SWIFT is massively scalable redundant storage system leverages in cloud solution. It integrated code from NASA Nebula as well as Rack Space cloud files platform. Nebula is a NASA's own storage tool and it is not related with Open Nebula.

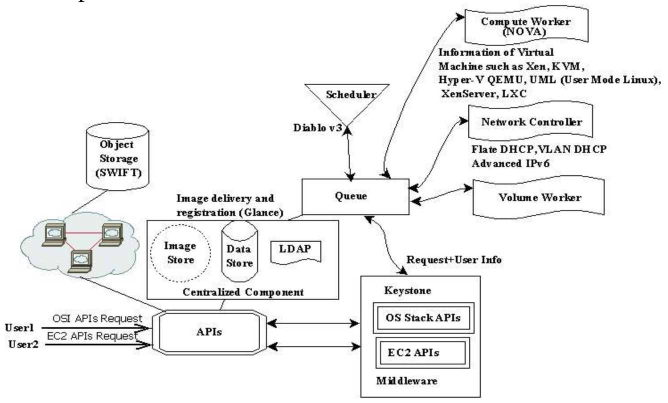

Fig. 4. Structure of open stack

It is defined as *Open Stack's Storage = Open Nebula Vs Nebula + Cloud Files*. Glance, which provide storage and retrieval of virtual machine images for Open Stack Nova component [9]. Fig. 4 represents the working of Open Stack.

2) Implementation

There are many ways to configure Nova, but the basics include the nova. conf file, setting up the database, and integrating networking. The various steps required to install Open Stack Included *setting up a volume group, setting up user, register an image, starting an instance and so on.*

Setting up a volume group: User will need to create a volume group for *nova-volume.*

Setting up user: The *nova.sh* script useful for the Nova administrator user, which created for user and an image already registered. For network configuration following commands are used in OpenStack.

```
user@user1:~$ sudo nova-manage network create
novanetwork   10.0.0.0/8 1 64
```

The *IP* address is the *cidr* notation for net mask, such as

192.168.1.0/24. Value one is the total number of networks user wants made, and the *64* values is the number of *IP* addresses in each network. To creating *nova* administrator user can use the following command.

```
user@user1:~$ sudo nova-manage user admin
openstackuser
```

For creating project for the user, now user can use the following command.

```
user@user1:~$ sudo nova-manage project create XYZ
openstackuser
```

User can use following command for accessing private key.

```
user@user1:~$ export EC2_ACCESS_KEY= 4e6498a2-
???-??? -??? -7d1333t97fdfgh
user@user1:~$ exportEC2_SECRET_KEY=0a520304-???-
??? -??? - 40sp34k05bggh
```

Now, user can download credentials for user/project and unzip the source credential. User has following command.

```
user@user1:~$ sudo nova-manage project zipfile XYZ
openstackuser
user@user1:~$ unzip nova.zip
```

Register an image: Before going to use instance from images, first user have to register those images. There are several sources for cloud-ready to get images from *tarballs* such as *Ubuntu Server 10.04 LTS, Ubuntu Server 10.10* and *ttylinux* test image (this image generally preferred in OpenNebula).

User can use *wget* for the available image or in Linux you can directly download the software from repository.

```
user@user1: ~$ wget http: //uec-images.ubuntu.com
/releases /10.04 / release / ubuntu- 10.04- server- uec-
amd64.tar.gz
```

User can use *uec-publish-tarball* from the *cloud-utils* package to register the *tarball* in one step.

```
user@user1:~$uec-publish-tarballubuntu-10.04-server-
uec-amd64.tar.gz mybucket
```

This will register user image in the *mybucket* bucket. The tool will output 3 references: *emi, eri* and *eki*. User only needs to use the *emi* value.

Starting an instance: The following commands is used to generate and register a *SSH* keypair for use in accessing the instances.

```
user@user1:~$ euca-add-keypair mykey > mykey.priv
user@user1:~$ chmod 600 mykey.priv
```

To run an instance using the *keypair* and *IDs* that we previously created we used the command.

```
user@user1:~$euca-run-instancesami-g06qbntt-k mykey-
t m1.tiny
```

Command repeatedly to watch as the scheduler launches, and completes booting your instance.

```
user@user1:~$ euca-describe-instances
```

Authorize *SSH* connections to the instance and connect: commands to *SSH* to the instance using your private key.

```
user@user1:~$ euca-authorize -P tcp -p 22 default
user@user1:~$ ssh -i mykey.priv root@ IP-adddress
```

After authorizing the *SSH* connection, now user can use the *VM*.

III. Comparison

Organizations can set up their own cloud computing plan to fulfill business requirements. There are several clouds computing tools and technologies; most of these tools based on open source environment and each have its own characteristics and advantages. There are various parameters such as cloud types, compatibility, deployment strategies, scalability, hypervisor support, security management, scheduling and policy algorithm, web interface, supported type, supported language, reliability, SSH key management, disk allocation, service type and latest release name. In this paper, Eucalyptus, Open Nebula, Nimbus and Open Stack are compared with respect to above mention parameters. The comparison of several cloud computing tools and technologies are represented in Table I.

Open Nebula and Open Stack support public, private and hybrid whereas Eucalyptus supports only private cloud. Nimbus supports public and private cloud. Eucalyptus provides storage compatibility with Elastic CC S3. Open Nebula supports open multiplatform, Nimbus and Open Stack compatible with Amazon EC2. The mention tools support KVM. Scheduling and policy algorithm, different scheduling algorithm works with different tools such as Eucalyptus Supports round robin and GREEDY, Open Nebula Supports rank matchmaker, packing, striping, load-aware and affinity-aware.

Storage management provides storage capability of Eucalyptus through walrus, Open Nebula and Open Stack through Amazon S3 and so on. Suggested configuration requirement that the suggested and minimum hardware and software requirements for deployment. Other criteria of comparison are also included such as, latest release name (latest release name), developer (developer of tool), reliability (tool is reliable or not) and structure (structure of the tool) etc.

IV. Discussion

Each cloud has its own characteristics that can be used according to the needs of users. OpenNebula provides virtualization platform to a pool of resources, decoupling the server not only from the physical infrastructure but also from the physical location. It contains OpenNebula controller to manage all nodes in the cloud, provides customizable option as per the requirement, suitable for smaller and private companies. Also it provides a greater level of centralization and customizability for end-users. It exposes more of the underlying features of libvirt to cloud users and administrators. It provides customization suitable for researchers of computer science that desire to experiment with combining cloud systems with other technologies, such as SGE or Condor, enables switching nearly every component, including the underlying file system, the DHCP, the front-end offers the best solution for communication with and controlling the VM layer of a cloud environment, deals with almost any data center combination of hardware and software, support VMware virtualization solutions apart from Xen and KVM, deploys and manages virtual machines individually or in groups on private resources or external public clouds, support for multiple types of users. In spite of advantages, OpenNebula has no control over hacking, capability to integrate into an existing authentication and authorization system.

Eucalyptus enables customers to create an on-site cloud infrastructure quickly and easily, emulates Amazon AWS, manages critical data behind the firewall and places non-sensitive data in public clouds. Also it supports for multiple hypervisor technology within the same cloud, enables the creation of private clouds that can interface with Amazon WS API. It separates user and admin space. Eucalyptus can easily be deployed on all types of inheritance hardware and software. It does not require any additional capital expense. It is easy to manage the underlying virtualized computation and storage. It has simple, flexible and modular architecture, which enables organizations to build their own clouds, web service API compatible with the EC2. API allows easy migration between a private and public cloud. However, if user wants to gain access to the full-feature set, users are required to buy a commercial license due to heavyweight Java implementation, lack of single sign-on support for its web interface. Eucalyptus VM placement algorithms are less powerful than OpenNebula's algorithms, as it does not allow any sort of expression of VM priority.

Nimbus pays the most attention to capacity allocation and capacity overflow. It is the only EC2 compatible open source IaaS implementation with support for spot pricing, which allows providers to leverage their better cloud cycles. It provides flexible tool for investigating and fine-tuning relationships between various aspects of resource utilization, energy and cost savings. Nimbus creates random address and it includes customization, provides support for KVM hypervisors and It is suited for small and mid-sized enterprises, flexible for working with various types of virtual networks. In spite of advantages, it is customizable for administrators and not for the user.

V. Conclusion and Future Work

This paper focuses on the architecture and implementation issues of Eucalyptus, Open Nebula, Open Stack and Nimbus. These tools may not emphasis on long-term support because of the version and OS dependency. Tools must be independent from the version and OS that can be used in the organization for a longer period of time. Scheduling algorithms play an important role in the deployment of cloud tools. Performance of cloud tools will depend upon the efficiency of scheduling algorithm used in the tools. The future work will focus on analyzing the unlike scheduling algorithm used by different cloud computing tools.

References

[1] V. A. Konstantinou *et al*., "An architecture for virtual solution composition and deployment in infrastructure clouds," *Technology in Distributed Computing*, pp. 9-18, 2009.

[2] R. Buyya *et al*., "Market-oriented cloud computing: vision, hype, and reality for delivering IT services as computing utilities," *Co RR*, pp. 599-616, 2008.

[3] J. Peng *et al*., "Comparison of several cloud computing platforms," *Information Science and Engineering (ISISE)*, vol. 15, pp. 23-27, 2010.

[4] L. Ramakrishnan et al., "Magellan: experiences from a science cloud," in *Proc. the 2nd international workshop on Scientific cloud computing*, pp. 49-58, 2010.

[5] D. Cerbelaud *et al*., "Opening the clouds: qualitative overview of the state-of-the-art open source VM-based cloud management platforms," in *Proc. 10th International Conference on Middle ware*, Urbana, Illinois, 2009.

[6] P. Sempolinski *et al*., "A Comparison and critique of eucalyptus, open nebula and nimbus," in *Proc. IEEE International Conference on Cloud Computing Technology and Science*, pp. 417-426, 2010.

[7] Amazon EC2. [Online]. Available: http: // www.amazon.com/ec2

[8] Open Nebula Incorporation Tool. [Online]. Available: http://opennebula.org

[9] Open Stack Deployment. [Online]. Available: http://wiki.openstack.org.

[10] Eucalyptus Deployment. [Online]. Available: http://open.eucalyptus.com.

[11] Nimbus Deployment. [Online]. Available: http://www.nimbusproject.org

Nitin Nagar has received master degree in computer applications from Devi Ahilya University, Indore and perusing PhD degree from Devi Ahilya University Indore, India. Presently he is the lecture at international institute of professional studies, Devi Ahilya University Indore, India. He is having more than 5 years of teaching and 3 years of research experience.

His areas of research are Cloud Computing, Advanced Database Management System, and Distributed Computing.

Ugrasen Suman has received master degree in computer applications from Rani Durgawati University Jabalpur and PhD degree in computer science from Devi Ahilya University Indore, India. Presently he is a reader at School of computer science & information technology, Devi Ahilya University, Indore, India. He is having more than 12 years of teaching and research experience.

His areas of research are software engineering, knowledge management & mining, software reuse, software maintenance & reengineering and service oriented computing. He has been guided Two PhD Scholar, Four PG research scholars and more than 50 PG projects. Currently he is guiding Eight PhD Scholars and Three PG projects. He has published more than 55 research papers in National & International Journals/ Conferences. He is also working as Dy. Coordinator on a UGC-SAP research project of Data Mining and Software Engineering.

He is a member of IEEE, IEEE-CS, Senior Member of IACSIT, Life Member of CSI and IAENG.