ISSN No. 0976-5697

Volume 8, No. 1, Jan-Feb 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Cloud Computing: A Perspective Approach

Sharda Rani
Assistant Professor
Deptt of Computer Science
R.K.S.D. College, Kaithal, India

Abstract- Cloud service models are infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Network as a Service (NaaS). Cloud deployment models are public cloud, private cloud, community cloud, and hybrid cloud. Cloud enables use of resources as per payment. It uses several technologies such as virtualization, multi-tenancy, and webServices. OpenStack is open source software to implement private and public cloud. It provides IaaS. This paper is a step towards alleviating efforts of setting up a private cloud. This OpenStack based implementation uses hardware assisted virtualization, logical volume management, and network interface card in promiscuous mode. Hardware assisted virtualization offers better performance. Logical volume management provides flexibility to file system expansion. Keystone, glance, cinder, quantum, and horizon services are grouped for one node, named controller node.

Keyword: Cloud infrastructure, virtualization, openstack, Service Models of Cloud.

I. INTRODUCTION

Cloud derives its name from the cloud shaped symbol representing Internet, as it is used as an abstraction for its complex infrastructure .Cloud computing is the use of the pooled computing resources accessible over Internet. Computing resources can be hardware or software.. It provides services as per requirement. It allows user to customize, configure, and deploy cloud services. It offers services as per payment. Cloud provides resources over Internet using virtualization technology, multi-tenancy, web services, etc. Virtualization provides abstraction of independent hardware access to each virtual machine. Multi-tenancy allows the same software platform to be shared by multiple applications. Multi-tenancy is important for developing software as a service application. Applications communicate over the Internet using web services [1].

A. Cloud Infrastructure

1) Deployment Models of Cloud

There are four deployment models of cloud.

a) Public Cloud: Public cloud makes services (such as computing, storage, application, etc.) available to general public. These services may be free or offered as payment as per usage. Major public cloud providers are Amazon, Google, Microsoft, etc.

b) Private Cloud: Private cloud is a cloud infrastructure operated only for a single organization. It is not available to general public.

c) Community Cloud: Community cloud shared infrastructure between several organization with common concerns such as compliance, jurisdiction, etc.

d) Hybrid Cloud: Hybrid cloud is a combination of two or more clouds (public, private, or community).

2) Service Models of Cloud

These are four service models of cloud: Infrastructure as a Service, Platform as a Service, Software as a Service, and Network as a Service. Figure 1 demonstrates the abstraction level of services. Software as a service is taken place at the top. From top to bottom services are more fine grained i.e., more access control to the resources [2].

a) Infrastructure as a Service (IaaS): Cloud provider offer computers as virtual machines, and other resources. Virtual machines are run as a guest by a hypervisor such as Xen or KVM. Other resources in IaaS could include images in a virtual machine image library, raw (block storage), file based storage, firewalls, load balances, IP addresses, virtual local area network (VLANs) and software bundles.IaaS cloud providers supply these resources on demand from their data centers. For wide area connectivity, the Internet can be used. Cloud provides a hosting environment that does not limit an application to a specific set of resources. To deploy their applications, cloud user install operating system image on the machine as well as their application software. In this model, cloud user is responsible for maintaining the operating system and application software.

b) Platform as a Service (PaaS): Cloud providers provide a computing platform typically including operating system, programming language execution environment (such as Java, Python, Go), database, and web server. Application developers can develop and run their software on a cloud platform. Open source implementations for PaaS are cloudfoundry, openshift origin.

c) Software as Service: In this model, cloud providers install and operate application software in the cloud and cloud users access the software from browser/client interface. Some cloud applications support specific client software dedicated to these applications (e.g., virtual desktop client, email client, etc.).Elasticity makes a cloud application different from other applications. This can be achieved by cloning tasks onto multiple virtual machines at run time. To accommodate a large number of cloud users, cloud applications can be multi-tenant.

d) Network as a Service: Cloud service where the capability provided to the cloud service user is to use network connectivity services. NaaS involves the optimization of resource allocations by considering network and computing resources. NaaS services include flexible and extended VPN, and bandwidth on demand.

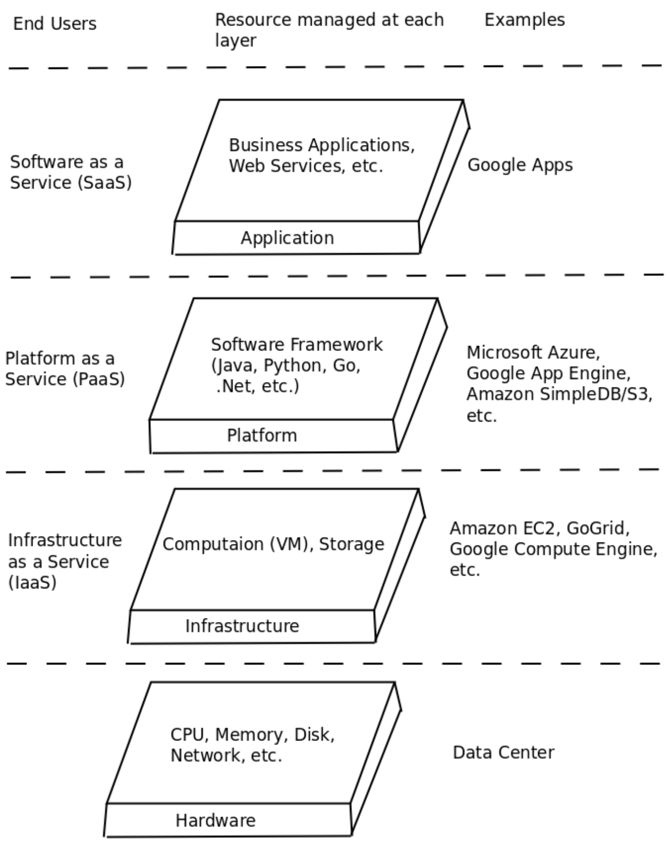

Figure 1: Cloud service models

II. RELATED WORK

The concept similar to that of cloud computing has evolved in decade of 1950s. Large scale mainframe was available and accessible via thin client/terminal. To utilize resources of mainframe, it allowed sharing the computer from multiple terminals. John McCarthy [3] speculated in 1960s that computation some day be organized as a public utility. As per [4], Amazon contended a key role in development of cloud computing. Amazon launched "Amazon Web Service" in 2006. In 2008, Eucalyptus became the first open source software for building Amazon web service compatible private cloud. OpenNebula is also a open source software for building cloud, released in 2008. In 2011, IBM announced the IBM SmartCloud.

A. *Open Source Implementation of Cloud*

1) OpenNebula [5]: is open source software to provide infrastructure as a service. It provides many interfaces to interact with facility offered to manage physical and virtual resources. This includes features for integration, management, scalability, and security. It provides interfaces like a Unix-like command line interface and the powerful Sunstone GUI. It provides image catalog, network catalog, and virtual resource control and monitoring.

2) Eucalyptus [6]: is an open source software and compatible with AmazonWeb Service (AWS). Cloud user can move instances between Amazon cloud and Eucalyptus cloud. Eucalyptus comprised of six components: cloud controller, walrus, cluster controller, storage Controller, VMware broker, and node controller. In March 2012, AmazonWeb Service and Eucalyptus announced details of the compatibility between AWS and Eucalyptus.

3) LingCloud [7]: is open source software developed by Institute of Computing Technology, Chinese Academy of Sciences. LingCloud can be used to build private cloud and manage the computing infrastructure. LingCloud uses the Xen based virtualization. The main components of LingCloud are molva and portal.

4) Nimbus [8]: is open source software to implement infrastructure as a service of cloud. It supports both Xen and kernel based virtual machine. The nimbus workspace service provides an implementation of compute cloud allowing users to lease computational resources. Cumulus provides an implementation of quota based storage cloud. Cumulus allows providers to configure multiple storage cloud implementations.

5) OpenQRM [9]: is open source software to implement IaaS. It supports Xen, VMware, KVM, and Citrix XenServer hypervisor. OpenQRM supports P2V (physical to virtual), V2P (virtual to physical), and V2V (virtual to virtual) migration. It supports multiple storage services such as NFS, iSCSI, ZFS, etc. OpenQRM has support for different distributions like Debian, Ubuntu, and CentOS.

6) OpenShift [10]: provides platform as a service provided by Red Hat. OpenShift has support for Ruby, Node.js, Python, PHP, Perl, and Java. It is built on open source technologies, and on foundation of Red Hat Enterprise Linux (RHEL).

7) OwnCloud [11]: provides location independent storage. OwnCloud can be accessed by web interface. With ownCloud, one can share file and directories. On android one may also create, download, edit, and upload any file.

8) Synnefo [12]: is an open source project for IaaS. Synnefo is compatible with the Open-Stack APIs. It uses Google Ganeti for the low level VM management.

9) Tsuru [13]: provides platform as a service. It is an open source software and developed by globo.com in 2012.

10) Scalr [14]: is an open source cloud computing platform for managing Amazon EC2.

11) Cloud Foundry [15]: is open source software to provide platform as a service developed by VMware. It is written in Ruby. Cloud Foundry support RabbitMQ, PostgreSQL, MySQL, MongoDB, Redies, etc. It supports Java, Ruby, Node.js, and Scala.

12) AppScale [16]: provides platform as a service. It supports Xen, KVM, Google Compute Engine, Amazon EC2, RackSpace, OpenStack, and Eucalyptus. It supports Python, Go, and Java. It is developed by AppScale Systems.

III. PROPOSED APPROACH

A. *OpenStack*

OpenStack is founded in 2010 by NASA and Rackspace, and now governed by the OpenStack Foundation. OpenStack project provides infrastructure as a service. More than 150 companies joined the project which is Intel, AMD, Canonical, SUSE Linux, Cisco, Dell, etc.OpenStack has modular architecture. It comprises of multiple Linux services. In Linux, a service is also known as a daemon. A service is a single program that runs in the background and listens on a port to react to service requests. OpenStack services are implemented by multiple Linux services. For example, nova- compute and nova-scheduler are two of the Linux services that implement the compute service. OpenStack also depends on several third-party services, such as database (e.g., MySQL, PostgreSQL, or SQLite) and messaging service (e.g., RabbitMQ, Qpid, etc.). Storage solutions are divided into three categories: object storage, block storage, and file storage. Some storage solutions support multiple categories. For example, NexentaS-tor supports both block storage and file storage; GlusterFS supports file storage and object storage, and Ceph Storage supports object storage, block storage, and file storage [17].In object storage, files are accessed through an HTTP interface, typically with a REST API. All data access is done at the user level and operating system is unaware of the presence of the remote storage system. User access and modify files by making HTTP requests. OpenStack image service can be configured to use the object storage service as a back-end. Object storage provides an HTTP interface. So it can be used as content delivery network (CDN) for hosting static web content (e.g. images, media files, etc.). Amazon S3, Rackspace Cloud Files, and Ceph Storage provide object storage [18].

In block storage, files are accessed through a low -level computer bus interface such as SCSI or ATA. User access data through the operating system at the device level by mounting the remote device in a similar manner as a local, physical disk (e.g., using the“mount” command in Linux). In OpenStack, the cinder volume service provides the block storage. Cinder uses iSCSI to access remote data as a SCSI disk that is attached to the network. Because the data is accessed as a physical device, the end user is responsible for creating partitions and formatting the disk device. In OpenStack block storage cannot be used to share data across virtual machine instances concurrently [18].

In file storage, files are accessed through a distributed file system protocol. User access data through the operating system at the file system level by mounting a remote file system. Examples of file storage include NFS and GlusterFS. The operating system needs to have the appropriate client software installed to be able to access the remote file system. Currently, OpenStack does not have any support for file storage. However, there is a Gluster storage connector for OpenStack that enables the use of the GlusterFS file system [18].

OpenStack is a toolkit for building a cloud environment by assembling together existing Linux technologies. Cloud controller provides all functionality of the cloud except actually hosting virtual machines or providing network services. This server hosts the image service, block storage service, identity service, and dashboard. It also run portions of compute service such as the API server, the scheduler, conductor, console authenticator, and VNC service. Finally, it hosts the API endpoint for the OpenStack network service. Network controller provides the network services such as DHCP, layer 2 switching, layer 3 routing. Compute node runs the OpenStack compute service as well as the OpenStack network service agent. This server also manages hypervisor such as KVM or Xen. This server hosts the actual virtual machines.

IV. IMPLEMENTATION

To build a cloud, open source services are stitched together. Apache web server, novnc(web based VNC client), and memcache are stitched together to provide web based interface to user. RabbitMQ is used for messaging service. Libvirt and kvm are used for virtualization. MySQL is used for databases. Logical volume manager and iSCSI are stitched together for storage service. Iptables, dnsmasq, and Linux bridging are stitched to provide networking service.

V. CONCLUSION AND FUTURE WORK

Resources such as hardware or software can be delivered over a network. These resources can be provided using collection of services or technologies (e.g., virtualization, multitenancy, web services, etc.). OpenStack is one of open source implementation of cloud which stitches open source services and technologies. Execution environment can be provided over network, where one can upload their application and execute. One can execute one instance of application and provides services to multiple customers. Users need not to despair about maintenance of machine and software application. Resources can be delivered as per requirement.

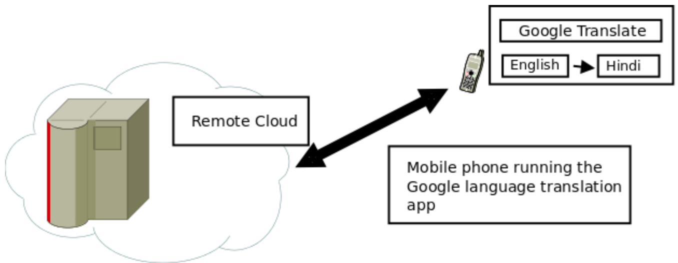

Figure 2: A remote cloud server providing services to mobile devices through the Internet

A. *Future Direction of Work*

Mobile device has scarce resources. Mobile device can use resources provided by cloud. So, how mobile device offloads its computation on the cloud. Section 5.2.1 describes architecture for mobile cloud computing.

1) *Mobile Cloud Computing Architecture*

There are different architectures for the mobile cloud computing. First, in which mobile device is directly connected to the cloud via Internet as displayed in figure 2. Second, in which other mobile devices act as a resource providers of the cloud as displayed in figure 3. Thus, resources of mobile devices in the local area and other stationary devices will be utilized [20]. Third, in which the cloudlet concept introduced by the M. Satyanarayanan [21]. As shown in figure 4, in this approach mobile devices offload their computation to the Cloudlet. Cloudlet is defined as a cluster of several commodity computers. Cloudlet is connected to the remote cloud. This cloudlet can be situated at the coffee shops or other dense populated area, so that mobile devices can connect to the cloudlet instead of a remote cloud.

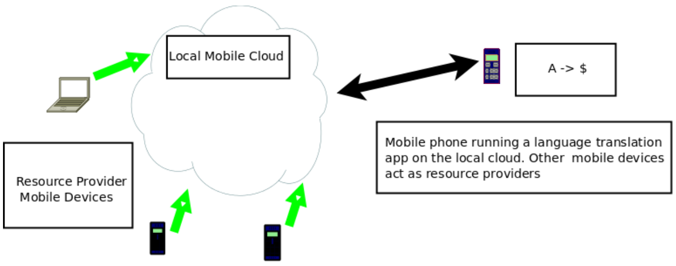

Figure 3: A virtual resource cloud comprised of mobile devices

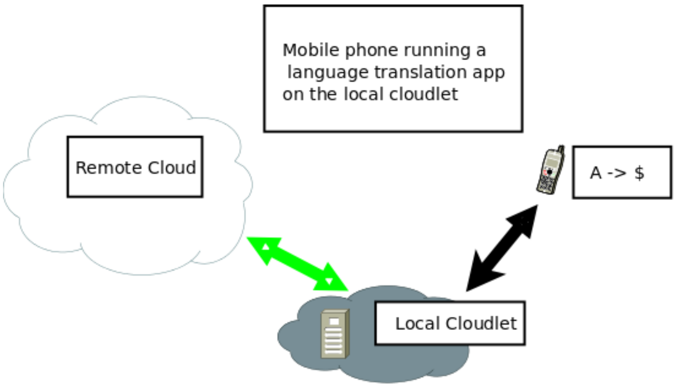

Figure 4: A cloudlet is used to make less severe latency and bandwidth issues and benefiting its resources.

VI. REFERENCES

[1] Gautam Shroff, "Enterprise Cloud Computing," Cambridge University Press.

[2] http://en.wikipedia.org/wiki/Cloud computing, Accessed on June 13, 2013.

[3] http://docs.openstack.org, Accessed on June 13, 2013.

[4] en.wikipedia.org/wiki/AppScale, Accessed on June 13, 2013.

[5] Niroshinie Fernando, SengW. Loke,Wenny Rahayu, "Mobile Cloud Computing: A

[6] survey," Future Generation Computer System 29(2013) 84-106, June 2012.

[7] M. Satyanarayanan, P. Bhal, R. Caceres, N. Davies, "The case for VM-based cloudlet in mobile computing," pp. 14–23, IEEE Pervasive Computing, 2009.

[8] Paramvir Bahl, Richard Y. Han, Li Erran Li, Mahadev Satyanarayanan, "Advancing the State of Mobile Cloud Computing," Proceedings of the third ACM workshop on Mobile cloud computing and services, June 2012.

[9] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce," Masters Thesis, Carnegie Mellon University, 2009.

[10] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, Paramvir Bhal, "MAUI: Making Smartphones Last Longer with Code Offload," Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, June 2010.

[11] Byung-Gon Chun, Sungwan Ihm, Petros Maniatis, Mayur Naik, Ashwin Patti,"CloneCloud: Elastic Execution between Mobile Device and Cloud," pp. "301–314", In EuroSys' 11: Proceedings of sixth conference on computer system, ACM,2011.

[12] D. Huang, X. Zhang, M. Kang, J. Leo, "Mobicloud: Buelding Secure Cloud Framework for Mobile Computing and Communication," Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering, 2010.

[13] Mads Daro Kristensen, "Scavenger: Transparent Development of Efficient Cyber Foraging Applications," Proceedings of the IEEE International Conference, 2009.

[14] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," vol. 51, Communications of ACM, Junary 2008.

[15] M. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: Enabling Interactive Perception Applications on Mobile Devices," Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM, Jun 2011.