International Journal of Computational Biology
ISSN: 2229-6700 & E-ISSN:2229-6719, Volume 3, Issue 1, 2012, pp.-44-47.
Available online at http://www.bioinfo.in/contents.php?id=96

THE APPRAISAL AND JUDGMENT OF NIMBUS, OPEN NEBULA AND EUCALYPTUS

NILESH MANGTANI* AND SUKHADA BHINGARKAR

MITCOE, Pune, MS, India.
*Corresponding Author: Email- nileshmangtani2306@gmail.com

Received: February 21, 2012; Accepted: March 15, 2012

Abstract- Cloud computing has been one of the hot research topics. It is the development of grid computing, parallel computing and distributed computing. A beginner or a cloud user with little knowledge it is still very difficult to choose the correct platform among the various platforms that are available now-a-days. There are various open source projects available that provides a substitute for the users that do not hope to use a commercially provided cloud. This paper mainly focuses on three main platforms for the cloud development- Nimbus, OpenNebula and Eucalyptus. Confidently this evaluation would help the beginners to choose the correct platform for their research .
Keywords- cloud computing, Nimbus, OpenNebula, Eucalyptus, cloud platforms, IaaS, PaaS, SaaS.

Citation: Nilesh Mangtani and Sukhada Bhingarkar (2012) The Appraisal and Judgment of Nimbus, Open Nebula and Eucalyptus. International Journal of Computational Biology, ISSN: 2229-6700 & E-ISSN:2229-6719, Volume 3, Issue 1, pp.-44-47.

Introduction

Cloud computing is originated from earlier large-scale distributed computing technology. Different paradigms have been developed for harnessing the computational power of large groups of processors for satisfying the high demand for computing resources for processing various jobs that required many cores. By providing different virtual machines to users we will shut in ourselves to consider the idea of IaaS. The word “cloud” in this state is meant to express the semi-ethereal nature of various VMs present in it.

In Parallel computing era, people strongly follow high-speed computing and use very many costly servers and also this investment can’t be reused again. In the cloud computing era, we don’t follow the use of expensive servers. It has a very important feature of on demand services. The user only requests the resources whenever needed from cloud computing centers and when the use is completed release the resources instantly. All this is done by the center of cloud computing, the user has the opportunity to take pleasure in high-performance computing because the cloud center can provide almost unlimited computing power, flexible computing and storage is also the important feature of the cloud computing flexibility. The cloud model mainly consists of five essential characteristics, three service models and four deployment models.

A. Essential Characteristics

1. Resource pooling- The resources are assigned and reassigned to the users as per the consumer demand. The users have no knowledge over the exact location of the resources that are provided.
2. On-demand self service- Without requiring human interaction with every service’s provider a consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically.
3. Rapid elasticity- To the users the capabilities available for provisioning often appear to be infinite and can be purchased in any extent at any time.
4. Broad network access: All the services are available over the network and they can be accessed by any devices such as mobile phones, laptops etc.
5. Measured Service- Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

B. Service Models

1. Cloud Infrastructure as a Service (IaaS)- Infrastructure as a Service delivers a platform virtualization as a service. Here

the user is fully responsible to provide the operating system kernel, base OS software. Rather than purchasing data center space, servers, software's clients instead buy those resources as a fully outsourced service.

2. Cloud Platform as a Service (PaaS)- This kind of service provides an environment as a service. The consumer does not manage or control the underlying cloud infrastructure but has control over the deployed. Anyone can use the middleman's equipment to develop their own program and deliver it to many other users through internet and servers.
3. Cloud Software as a Service (SaaS)- This is as a "pay-as-you-go" type of model. The consumer accesses the applications through a web browser that are provided by the provider on the cloud infrastructure.

C. Deployment Models

1. Public Cloud- The public cloud is owned by an organization selling cloud services and all the customers share the computing resources provided by one service provider.
2. Private Cloud- Private Cloud is managed by a single organization or a third party and all the computing resources are used and controlled by a private enterprise only. The main advantage of this is that the security of data is under the control of enterprise only.
3. Hybrid Cloud- This is a composition of two or more clouds. It enables the enterprise to ask for the computing resources when peak workload occurs, and then return if no longer is needed.
4. Community Cloud- The cloud infrastructure is shared by several organizations. It supports a specific community that has shared the same policies, requirements, values, and concerns.

In this paper sections II, III and IV, describe Nimbus, OpenNebula and Eucalyptus respectively. The discussion about these three solutions is presented in Section V. Finally, Section VI summarizes our conclusions.

Nimbus

A. Architecture

Nimbus is a cloud computing solution providing IaaS. It permits users to build the required computing environment and lease remote resources through the deployment of virtual machines. Figure 1 shows that it includes many components such as gateway, agent, and clients and so on which are classified in three types.

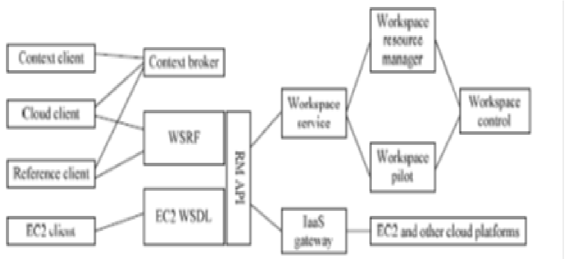

Fig. 1- Nimbus Architecture

First type is client-supported modules which are used to support all users of clouds. Context client module, cloud client module, reference client module and EC2 client module are all belong to this kind of component. The next type is service-supported modules. It mainly deals with all the different kinds of services for clouds. It includes context agent module, web service resource framework module, EC2 WSDL module and remote interface module. The last type is background resource management modules. This are essentially used to manage all kinds of physical resources on the cloud computing platform, including work service management module, IaaS gateway module, EC2 and other cloud platform support module, workspace pilot module, workspace resource management module and workspace controller.

B. Networking

The various VMs present in Nimbus have a compute node to run a DHCP server on it. This DHCP server is configured by Nimbus such that the IP address is assigned to the similar MAC address just at random formed. Nimbus has great interest in "One-click clusters" which are actually groups of VMs. They are formed by a virtual network between them. A "One-click cluster" is actually a static group of VMs. This avoids job scheduling in concept of cloud and provides the leases to VMs.

Open Nebula

A. Architecture

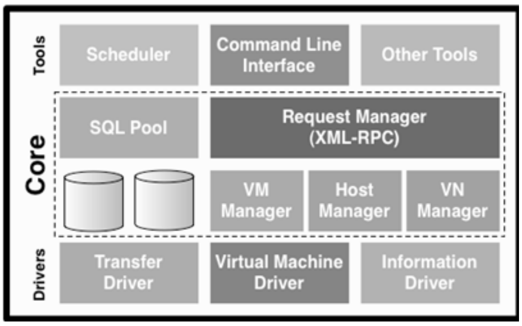

Fig. 2- Open Nebula Architecture

Open Nebula was firstly successfully implemented in central library in Florence. It was published by Ignacio M. Llorente and Rubén S. Montera. It sets user's clusters so flexible that it is possible for virtual infrastructure to adapt change of the service load. It has been designed in such a way that it allows integration with many different hypervisors and environments. There is a front-end that executes all the process in OpenNebula while the cluster nodes provide the resources that are needed by VM. There is at least one physical network joining all the cluster nodes with the front-end. As already mentioned, OpenNebula converts a physical cluster into a flexible, virtual infrastructure, which can dynamically adjust the resources required. Physically it consists of three layers: tools, core, and drivers as shown in Figure 2.

The Tools layer contains functions for the administrators and users. One component is the Command Line Interface that is used by administrators. In this, administrators access it through command line interface while the user manages VM with the help of web services interface. Second layer is the core layer that contains components responsible for handling client requests and control resources. The VN Manager manages virtual networks by keeping track of IP and MAC addresses and their association with VMs. The final layer is Drivers layer that supports the various platforms below it and it is formed by modules called as Drivers.

These Drivers are used to request services from external clouds like Amazon EC2 or ElasticHosts. These drivers deal with file transfers that are implemented by protocols like SSH.

B. Networking

There are virtual networks and VMs present in the OpenNebula. It manages the IP and MAC addresses between them. There are basically two types of virtual networks: Fixed Network and Ranged Network. Fixed Network uses a fixed set of IP and associated MAC addresses whereas the Ranged Network (Red LAN) is defined over a range of network addresses. VMs must be relevant to one Red Lan.

Eucalyptus

A. Architecture

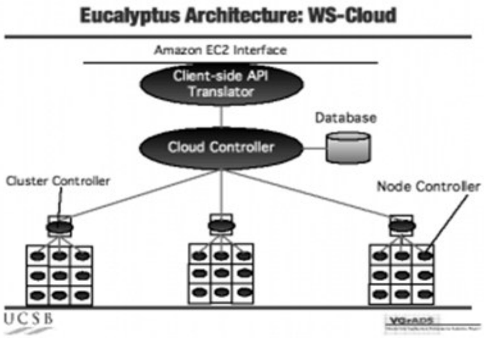

Fig. 3- Eucalyptus Architecture

The hierarchical design of EUCALYPTUS reflects common resource environment found in many academic settings. In EUCALYPTUS each high-level system component is implemented as a stand-alone Web Service. The benefits of this are: first each Web service exposes a well-defined language-agnostic API in the form of a WSDL document containing both operations that the service can perform and input/output data structures. Second, we can leverage existing Web-service features such as WSSecurity policies for secure communication between components. The high-level components are: *Cloud Controller (CLC)* is the entry point into cloud for all i.e. administrators, developers, managers and end-users. It is responsible for querying the node managers for information about resources, making high-level scheduling decisions, and implementing them by making requests to cluster controllers. *Cluster Controller (CC)* gathers information about set of Virtual Machines and schedules its executions on specific NC. It executes on any machine that has network connectivity to both nodes running NCs and to the machine running the CLC. It also manages the virtual instance network. *Node Controller* is executed on each and every node that is designated for hosting VM instances. Node Controller controls the execution, inspection, and terminating of VM instances. *Storage Controller* is a put/get storage service that provides a mechanism for storing and accessing the virtual machine and user data. It permits users to put, get, list buckets, create, delete objects and set access control policies.

B. Networking

Client accessed the VM through an external IP. VMs can exchange traffic together which are of the same owner in two types: internal and external. In order to make a separation between them in EUCALYPTUS two virtual network interfaces are present: public and private. The private interface is used for inter-VM communication across different subnets. It actually handles the situation where two VM instances are running inside separate private networks but both of them have to communicate with each other. These instances are connected via a process level implementation of the Ethernet protocol called as Virtual Distributed Ethernet.

Discussion

In this paper, we have compared three different open source platforms. The main aspects are main purpose, platform architecture, networking aspects. At the end of this discussion, some use scenarios for each platform are discussed.

Eucalyptus can offer the same service as Amazon as it is compatible with EC2 and S3. It provides the softness required to cover varying levels of demand of user which avoid the website from becoming overloaded. The best example where Eucalyptus can be used at its best is a place where dealing is done with large amounts of data or which need to carry out major calculations. Possible candidates here are scientific or research institutions, but also commercial enterprises in the financial sector or in the automotive industry.

Similar to Eucalyptus, the option of dynamically adjusting the resources required is a benefit of OpenNebula. Any institution wishing to improve its current network solution can use OpenNebula. The important reason behind this is that it is designed in such a way that it can be integrated into any existing solution. Installing OpenNebula is relatively straightforward because an installation guide illustrating the various installation options is made available on the website.

Table 1- Summary of Different Cloud Platforms

	Nimbus	OpenNebula	Eucalyptus
Cloud Character	Public	Private	Public
Language	Java, Python	Java	Java
Hypervisors supported	Python, Bash, Ebtables, Libvirt, KVM, Xen	Xen, VMWare	VMWare, Xen, KVM
Web Interface	EC2 WSDL, WSRF	Libvirt, EC2, OCCI API	Web Service
User interface	Web-Services, specifically: Nimbus Web	Command Line interface (CLI)	euca2ools (CLI)
Internal Security	Fairly tight, unless deploying a fully private cloud.	Looser, but can be made more tight if needed.	Tight. Root required for many things.
User Security	Users x509 credential is registered with cloud	User logs into head (unless optional front-end used)	Users are given custom credentials via a web interface
Network Issues	dhcpd on every node and Nimbus assigns MAC	Admin must set manually but has many options	dhcpd on cluster controller
Unique Features	Nimbus context broker	VM migration supported	User management web interface
Area of application	Research institutions	Large commercial companies and public institutions	Large commercial enterprises Research institutions

Nimbus looks toward the more cooperative scientific community that might be less interested in the technical internals of the system, but has broad customization requirements. It is a cloud computing solution which is mainly suited for use in scientific domain, because the platform requires severe user orientation. Moreover, compared with other platforms, a cloud can be adapted in a much more individual way, for example with regard to supporting different hypervisors or user interfaces.

Suppose we have a homogeneous pool of Xen or KVM hypervisors and want to offer a public cloud service to a community, more specifically offering a simple IaaS. In this case, the free version of Eucalyptus will fit very well. Now, consider that we have a pool of resources, possibly with heterogeneous virtualization platforms, and want to create a private/hybrid cloud over it. OpenNebula can easily support this. Also, suppose we do want to provide IaaS with more diversified and powerful functionalities, as virtual machine migration and a more useful resource allocation mechanism. Then OpenNebula can also provide this. Overall, OpenNebula seems to be more suited for research and experimental studies. Table 1 provides a short summary of cloud platforms compared here.

Conclusion

In this paper, we have presented a detailed description about three Open Source Platforms. We have tried to explain the different features of all the three platforms so that one can decide which platform should be used under what condition to maintain & deploy the cloud in an easy manner by understanding the different features between them.

References

[1] Garg C.S. and Huylebroeck J. (2009) 10*th ACM/IFIP/USENIX International Conference on Middleware*, (22).

[2] Endo P.T. Gon calves G.E., Kelner J. and Sadok D. (2010) *Brazilian Symposium on Computer Networks and Distributed Systems*.

[3] Lei Z., Zhang B., Zhang W., Li Q., Zhang X. and Peng J. (2009) *International Symposium on Information Science and Engineering*, 23-27.

[4] Rimal B.P., Choi E. and Lumb I. (2009) *Fifth International Joint Conference on INC, IMS and IDC*, 44-51.

[5] Peng J. (2009) *Second International Symposium on Information Science and Engineering*, 23-27.

[6] Eucalyptus (2011) *http://eucalyptus.com*.

[7] OpenNebula (2011) *http://opennebula.org*.

[8] Nimbus (2011) *http://nimbusproject.org*.