ANALYSIS OF CLOUD COMPUTING IN OPEN SOURCE PLATFORM

Fui Fui Wong[1, 2] and Cheng Xiang Tan[1]

[1]*Department of Computer Science and Technology, TongJi University, Shanghai, 201804, China*

[2]*School of Information Technology, Kuala Lumpur Infrastructure University College, Kajang, 43000, Malaysia*

Keywords: Open source cloud computing, Cloud computing, Eucalyptus, OpenNebula, Nimbus, abiCloud, openQRM, Xen cloud platform.

Abstract: There is an increasing interest in the use of open source cloud computing infrastructure platform by larger companies, academic researchers, scientists and service providers that can tailor the platform for specific users. The open source cloud computing has joined the mainstream where it is starting to offer companies a lot of flexibility and cost savings to combine public and private clouds. These tools could let companies to build and customize their own computing clouds to work alongside with more powerful commercial solutions. To understand these issues, this paper outlines an analysis study of open source cloud computing platform using six of the most advanced platforms (Eucalyptus, OpenNebula, Nimbus, abiCloud, openQRM and Xen Cloud Platform) in order to gauge whether its potential opportunities could outweigh their possible drawbacks.

1 INTRODUCTION

Cloud computing provides a valuable way for many business and government organizations to implement low cost, low power and high efficiency systems to deliver scalable infrastructure. It means that there will be no upfront investment in servers or software licensing. Instead of spending millions of dollars and months in building their own infrastructures, a paradigm shift is taking place to let smaller organizations tap on the infrastructure of giant organizations According to a recent Gartner research report, cloud computing is expected to be the top priorities for CIOs in the year 2010 as the businesses shift from a traditional in-house data center to a hosted solution.

However, there are growing concerns over the control ceded to large commercial cloud vendors come with drawbacks, including the lack of information privacy, security, and reliability concerns. To understand these issues, this article aims to take a closer look at the open source cloud computing platforms in order to gauge whether their potential opportunities outweigh their possible drawbacks.

2 EMERGING OPEN SOURCE CLOUD COMPUTING INFRASTRUCTURE PLATFORMS

Lately, the open source cloud computing has joined the mainstream where it is starting to offer companies a lot of flexibility and cost savings to combine public and private clouds . It has become very clear that open source cloud platform occupies a significant portion of cloud market. Thus, it is important to examine the characteristics and suitability of open source cloud computing platforms for the public understandings.

2.1 Eucalyptus Cloud Computing

Eucalyptus (2010e) is a Linux-based open source software developed by the Computer Science Department at the University of California, Santa Barbara as the research project to support the high performance distributed computing environment (Nurmi et al., 2009). Eucalyptus aims to implement on-premise and private clouds within an enterprise's existing IT infrastructure that is commonly referred as Infrastructure as a Service (IaaS). The most

Fui Wong F. and Xiang Tan C..
ANALYSIS OF CLOUD COMPUTING IN OPEN SOURCE PLATFORM.
DOI: 10.5220/0003382701580165
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 158-165
ISBN: 978-989-8425-52-2

remarkable feature of Eucalyptus is emulated many Amazon Elastic Compute Cloud (Amazon EC2) functionalities has attracted many researchers and IT professionals to use this free version of EC2. For example, Eucalyptus supports many EC2 functionalities like the EC2 interface, the Amazon Simple Storage Service (Amazon S3) interface, the Elastic Block Store (EBS), Zone, Elastic IP addresses, and security groups. The supported virtualization techniques are: Xen and Kernel-based Virtual Machine (KVM).

2.1.1 Eucalyptus Design

The Eucalyptus infrastructure consists of five main components, the Cloud Controller (CLC), the Cluster Controller (CC), the Node Controller (NC), the Storage Controller (SC) and Walrus (Nurmi et al., 2008). The deployment of the Eucalyptus is engineered to mimic the Amazon Web Services (AWS) as a result of supporting AWS Application Programming Interfaces (APIs) from Eucalyptus without modification. Figure 1 shows the Eucalyptus architecture.

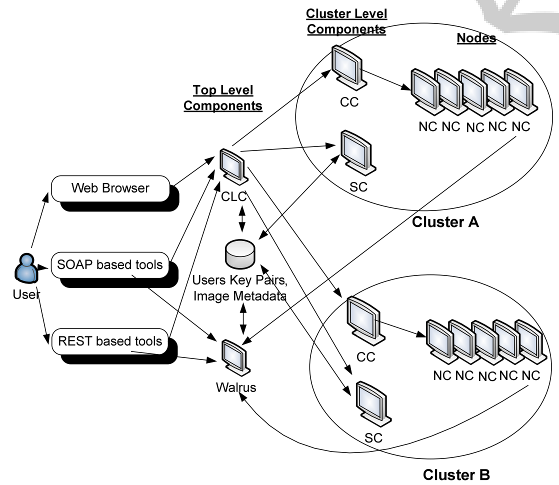

Figure 1: Eucalyptus environment.

The cloud controller (CLC) is a top-level component that aggregate resources from multiple clusters to accommodate the incoming user requests and authentication, resource allocation, persistent user and system data, manage and monitor the underlying virtualized resources. The Walrus is also a top level component in a cloud architecture that written in Java program to implement bucket-based storage through Amazon S3-compatible SOAP and REST interfaces.

The Cluster Controller (CC) is a cluster-level component that executes on a cluster front-end machine to handle cluster-level scheduling and network control. Each cluster requests a cluster controller (CC) to communicate with both the nodes running Node Controllers (NC) and to the machine installing the CLC. The Storage Controller (SC) is one of the cluster-level components that implement EBS-style block-accessed network storage. The SC emulates the functionality of Amazon Elastic Block Storage. It manages dynamic block devices (e.g. EBS) that Virtual Machines (VM) can use for persistent storage as well as supporting various storage systems (e.g. NFS, iSCSI, etc.)

Each physical machine in the cluster with a hypervisor will need a Node Controller (NC). The NCs control the execution, monitor the state information and termination of the VM instances on that particular machine in response to enquiries and control requests from the cluster controller (CC).

2.2 OpenNebula Cloud Computing

OpenNebula (ONE)(2010g) is an open source toolkit for cloud computing to build any type of Cloud deployment (e.g. Private, Hybrid and Public Clouds). Its origins are as a research project at Complutense University of Madrid in 2005. The aim of the OpenNebula is to produce a Virtual Infrastructure (VI) management solution with the flexible architecture, interfaces and components that fits into any existing data center and external clouds (Sotomayor et al., 2009) .

2.2.1 OpenNebula Design

The OpenNebula private cloud implements a classical cluster-based architecture, as shown in figure 2. It consists of a front-end cluster which also known as administration node, several cluster nodes where virtual machines (VM) will be installed, and an image repository that holds the base images of the VMs. Communications between front-end and cluster nodes use SSH.

The administration node executes on a front-end cluster machine, interconnect with a LAN to the cluster nodes that serve the user requests. As shown in the figure 2, the front-end machine is where OpenNebula is installed. OpenNebula is a Virtual Infrastructure (VI) manager that is executed and configured by a Front-End cluster. The OpenNebula internal architecture (see Figure 3) can be classified into three layers, the Tools, the Core and the Drivers dedicated to different aspects
of VI management (2010f).

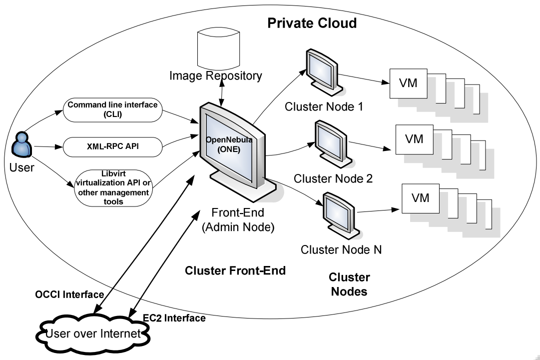

Figure 2: OpenNebula design.

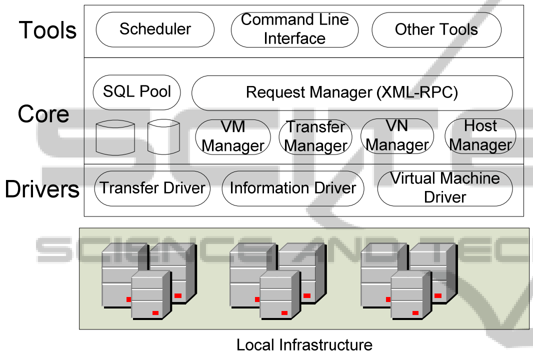

Figure 3: OpenNebula internal architecture.

To summarize, the tools layer consists of 3 components, Scheduler, Command Line Interface (CLI) and other third party tools developed using the interface given by the OpenNebula Core layer. The Command Line Interface allows the infrastructure administrators and users to submit, control and monitor the virtual machines; to add, delete and monitor the hosts, virtual networks and users respectively. The scheduler component is an independent entity in the OpenNebula architecture that is responsible for the assignment between the pending Virtual Machines and it is known as Hosts. The Haizea Lease Manager(2009a) can be used as a drop-in replacement for OpenNebula's scheduling daemon to support advanced reservation and preemption of resources and queuing of best effort requests (Sotomayor et al., 2008).

The Core layer encompasses several components to control and monitor virtual machines, virtual networks, storage and hosts through pluggable drivers. To perform VM's life cycle, the OpenNebula core orchestrates six different management areas: the request manager for answering client requests; the virtual machine manager for the management and monitoring of VMs; the transfer manager for transferring VM images among the cluster node, image repository and cluster front-end; the virtual network manager for creating of virtual network and handling of IP and MAC addresses; the host manager for managing and monitoring the physical hosts; the database for storing the persistent data needed in the management VMs.

Drivers are separate processes that communicate with the OpenNebula core.

2.3 Nimbus Cloud Computing

The Nimbus toolkit is an open source cloud computing that help to turn your existing infrastructure into an Infrastructure-as-a-service (IaaS) cloud with cloud-like interfaces. Its primary objective is to provide an IaaS-style cloud platform to the scientific community. The Nimbus v2.4 version eliminates the need for a separate Globus container installation and added significant amount of new improvements and features. It also leverages Amazon EC2 WSDLs, Amazon EC2 Query API and Grid community WSRF interfaces and offers self-configuring virtual cluster support. Virtualization implementation is based on Xen and KVM

2.3.1 Nimbus Design

The architecture of the Nimbus system is simple and flexible with a list of components to allow a user to dynamically deploy and manage VMs. The software framework is a highly modular cooperative set of web services that interoperate using standard communication protocols. Each Nimbus component is lightweight and self-contained that can be sanely replaced by other module in a variety of ways. The Nimbus 2.4 version contains a piece of software called Nimbus service will install on one service node per site and another piece of software known as Control agent to setup on Virtual machine monitor (VMM) nodes as shown in Figure 4. Clients interact with the service using X.509 credentials via multiple protocols. User can choose to launch VM via different clients interface.

The Nimbus service internal architecture (see Figure 5) encompasses several components specialized in different aspects of creating, controlling, and monitoring virtualized resources on an infrastructure-as-a-service cloud. The workspace service is a site manager which allows remote protocol frontends (for example, web service resource framework (WSRF), EC2 WSDL)) to deploy and manage VMs. The workspace resource manager executes VM deployment. The workspace pilot implements the integration with existing VMs with the job assigned.

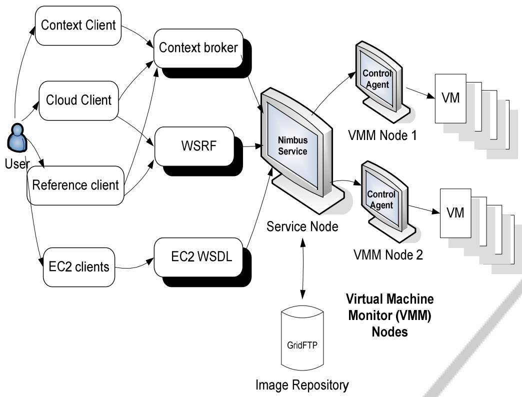

Figure 4: Nimbus design.

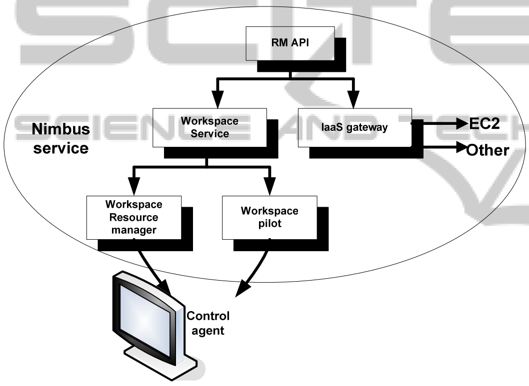

Figure 5: Nimbus service internal architecture.

2.4 abiCloud

Abicloud (2010a) has been developed as an open source cloud solution for implementing the creation and integral management of public and private cloud in heterogeneous environment. It was developed and maintained by Abiquo(2010c). AbiCloud is mainly active in Spain which contains three different versions (such as community, enterprise and xSP versions) to target the different needs of commercial organizations.

2.4.1 abiCloud Design

The AbiCloud design is primarily motivated by the concept of resource management where it applies the total separation of the physical infrastructure from the virtual application infrastructure. Therefore, AbiCloud is as scalability as a result of its loosely coupled modular system. The Abicloud (Figure 6) is composed of three main components (2010d). The abiCloud-Server is the core management software written in java that is running in the centralized servers store of the global business logic of the cloud platform. It acts as a front-end to communicate with the database, the rest of the modules in AbiCloud platform and extensible by adding more servers components and load balancer. The abicloud WS (Web Service) is responsible for the management and monitoring of Virtual Machines. The plug-in manager works with WMware ESX and ESXi, Microsoft Hyper-V, Virtual Box, Xen, KVM. The abiCloud VMS (virtual monitor system) is in charge of the virtual infrastructure for the correct deployment of virtual machines by keeping track of the events or states in the cloud.

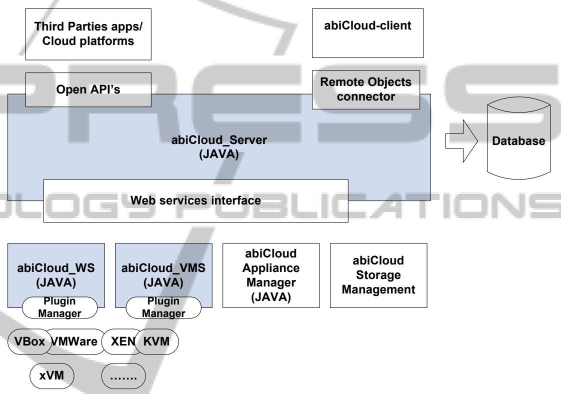

Figure 6: Platform architecture of abiCloud (2010d).

2.5 openQRM Cloud Computing

The openQRM (2010b) (open Qlusters Resource Manager) was initially developed by the Qlusters Company back in year 2001 as the commercial license software to manage the virtual machine deployment in data center. The openQRM 4.6 supports the Amazon EC2 adapter which allows for the creation of hybrid Cloud Computing, migration to other cloud provider such as Ecuclyptus or UEC and vice versa(2010h). The supported virtualization techniques are: VMware, Xen, KVM and Linux VServer.

2.5.1 openQRM Design

The platform design of the openQRM is primarily motivated by two engineering goals: high-availability and automatism. openQRM is based on a strictly pluggable architecture so that they can be selected and composed in a variety of ways. It has no feature of itself but all the features of openQRM are provided by its plugins. Indeed, the openQRM server is designed to just manage the plugins through its automated and generic interfaces without changing a single line of code in the base server (Rechenburg, 2010). The proven open-source and commercial third party plugins in openQRM perform the actual implementation which integrated into the openQRM server framework to communicate with the internal server-object to carry out the required action. The plugins tools are ranging from Server-management, Storage-management, deployment, provisioning, remote administration, local server integration, high availability, real-time monitoring and etc (figure 7).

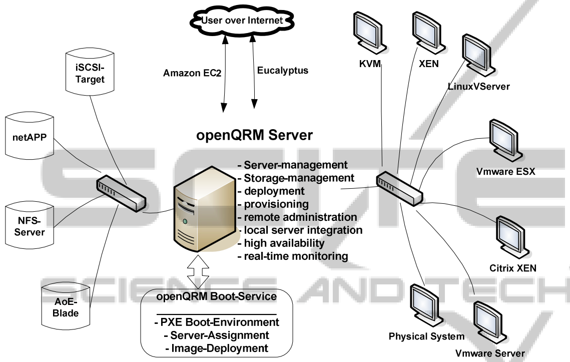

Figure 7: Platform architecture of openQRM server.

2.6 Xen Cloud Platform

Xen Cloud Platform (XCP) (2010i), a project initiated by Xen.org community. The most striking feature of Xen Cloud Platform is its infrastructure supports other popular open source projects, for example, Eucalyptus, OpenNebula, Nimbus, Convirture, OpenXenCenter and Xen VNC Proxy to leverage the Xen hypervisor on cloud computing environment (2009d). The release version of XCP 0.1 leverages other Xen tools, such as Latest Xen 3.4.1, Linux 2.6.27 Kernel, Window PV Drivers, Citrix XAPI enterprise-class management tool stack, basic SR-IOV Support. It has the particularity to address the need of ISVs and service provider by supporting a complete virtual infrastructure cloud service for multi-tenancy, SLA guarantees, security and network virtualization technologies (2010i).

2.6.1 Xen Cloud Platform Design

The Xen Cloud Platform is a server virtualization platform that consists of four main components as shown in figure 8 (2009b) that is dedicated entirely to the task of hosting VMs. The Xen hypervisor is installed between the server's hardware and operating system which provides an abstraction layer to run one or more guest operating systems. The control domain is a virtual machine that is responsible for controlling host hardware device and creating more guest domains. Storage manager (SMAPI) and XAPI management stack are components that running inside the control domain. To be a part of resource pool, a collection of Xen Cloud Platform host that group together into a single unit of administration which can host virtual machines across a cluster of machines (2009c).

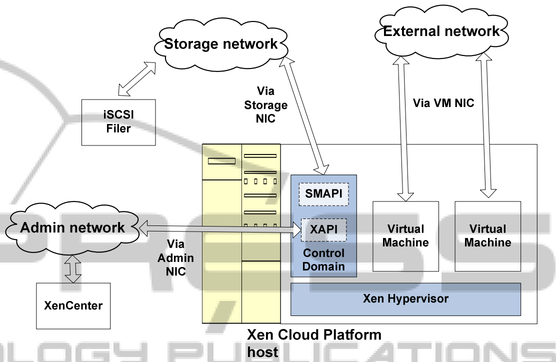

Figure 8: Xen cloud platform architecture.

3 COMPARISON OF THE 6 CLOUD INFRASTRUCTURE PLATFORMS

In this section we analyze six major open source cloud computing infrastructure players in this arena against 5 levers: scalability, high availability, interoperability, virtualization management and deployment management.

Scalability refers to the ability of a computer system to be scaled up easily to meet the increased demand in order to maintain performance levels as the number of concurrent requests raises. For instance, Eucalyptus does not support for auto scaling; OpenNebula allows dynamic resizing the physical infrastructure which makes it easily to expand or shrink according to the demand although it is not fully automated; Nimbus permits the client to specify the resource allocation during deployment; Abicloud allows to automatically modify soft limits to scale the infrastructure which let the users to control the provision of servers, storage, networks, virtual network devices and applications. OpenORM supports growth and auto-scaling through a power management plug-in. XenCloud anticipates the future expansion by manually adjusting the capacity through the workload balancing deployment.

Availability concerns may stop some companies from using cloud computing for transaction oriented applications due to the cloud computing has not always provided round-the-clock reliability. Looking at the current providers, Eucalyptus, Nimbus and AbiCloud are not supported for fault tolerance whereas no documentation can be found about this feature; OpenNebula using a persistent backend database to store their hosts and VM information through its repository; openQRM maintains a stand-by server for fault tolerance that allows groups of servers to use this single backup server to achieve the N to 1 failover concept; XenCloud allows administrator to restart the failed VMs on another host if high availability is activated on the resource pool.

Interoperability creates an open infrastructure / framework that let applications and data to seamlessly work together for the purpose for wider industry adoption of cloud computing technology. For example, Eucalyptus and openQRM support the Amazon EC2 API; openNebula has the capacity for integrating with other cloud computing tools like Eucalyptus, Amazon, ElasticHosts and Nimbus and supports open standard such as OGF's Open Cloud Compute Interface specification and Open Virtualization Format (OVF); Nimbus has an open API which enables the integration of other third party applications such as Amazon EC2 WSDLs, Amazon EC2 Query API and Grid community WSRF; abiCloud recommends the usage of open virtualization format between datacenters and integrate with the Sun's Open Cloud API; XenCloud will interact and integrate with other tools and products, like Eucalyptus, Convirture, OpenNebula, OpenXenCenter, Xen VNC Proxy, and Nimbus. It is also compatible with the Distributed Management Task Force (DMTF), CIM and Open Virtualization Format (OVF) standards.

Virtualization technology is a vital part of establishing platform leadership. It would be impressive if live migration of running virtual machines are able to be carried out from one physical server to another with zero downtime and undetectable to the users. However, this feature is yet to be a part of the cloud framework. For instance, Eucalyptus, Nimbus and abiCloud are not supported for managing machines on a cluster with features as live migration; OpenNebula supports live migration but only can be implemented with Shared FS; openQRM with remarkable solution for various migration scenarios such as Physical to Virtual, Virtual to Physical, Virtual to Virtual; XenCloud supports live migrations through its complete list of plug-in.

One aspect of the deployment management is the user interface. Graphical user interface to access the cloud can ease the deployment of cloud compared to the command line that acts as a barrier to entry. For instance, Eucalyptus has well integrated with third party API, with the use of graphical tools from RightScale's and amazon web interface, Nagios, Ganglia, Elasticfox and Hybridfox to ease the management of virtual machine instances; OpenNebula relies on the command line interface (CLI) for the deployment of cloud. Nimbus comes with a friendly web interface to manage administrative and user functions. AbiCloud has a straightforward and comprehensive one stop graphical user interface build with Adobe Flex that decreases the effort of the platform deployment. OpenQRM provides a user-friendly web interface to monitor the status and performance of its managed subsystems. XenCloud has a web interface that used to manage the VMs and perform live migrations via a web browser.

Table 1: Summary of comparisons among cloud infrastructure platforms.

Criterion	Eucalyptus	OpenNebula
Auto Scaling	No	No
Fault Tolerance	No	Yes (use persistent database backend to store hosts, networks and virtual machines)
Interoperability	Amazon EC2/S3/EB, Rackspace, GoGrid.	Amazon EC2, ElasticHost, GoGrid, Rackspace, Terremark, RimuHosting Eucalyptus, Nimbus
Live migration	No	Yes
User interface	CLI	Unix-like command line interface

Different strategies are employed by the open source infrastructure platform vendors. Clearly there is no single vendor which is good in all identified criterions. We cannot say one vendor is better than the others. However, enterprises need to know their key concerns when moving their data center into the cloud.

Table 2: Summary of comparisons among cloud infrastructure platforms (contd.).

Criterion	Nimbus	Abiquo
Auto Scaling	No	No
Fault Tolerance	No	No
Interoperability	Amazon EC2	Sun's Open Cloud API
Live migration	No	No
User interface	web interface	Graphical User Interface

Table 3: Summary of comparisons among cloud infrastructure platforms (contd.).

Criterion	openQRM	Xen Cloud Platform
Auto Scaling	Yes (via its power management plug-in)	No
Fault Tolerance	Yes (with fault-tolerance and fail-over capabilities to all applications)	Yes (with the ability to restart the failed VMs on other machine automatically)
Interoperability	Amazon EC2, Eucalyptus, UEC	Eucalyptus, Convirture, OpenNebula, OpenXenCenter, Xen VNC Proxy, Nimbus
Live migration	Yes(P2V, V2P, V2V)	Yes
User interface	Web Interface	Web Interface

4 CONCLUSIONS

Along the way, the open source movement has touched almost every sphere of cloud computing that we know today. It is good to see open source becomes one of the key drivers competing in the cloud computing world to provide more flexible and rich capabilities to cloud providers. The use of the open source platform as a viable alternative to proprietary software is of particular interest to the organizations today. However, presently open source cloud computing is relatively new and mostly still in the continuous development stage and yet to reach the maturity level. The work presented in this article will need updates as new releases are constantly emerging. Looking forward to see more innovative features in the open source platforms that eventually would benefits the human societies in the same way as what is Linux has done to the world since it has been developed. We believe that cloud computing will change and evolve our daily life.

REFERENCES

History (Online). Available: http://www.opennebula.org/about:history (Accessed 20/4/2010).

Nimbus (Online). Available: http://www.nimbusproject.org/ (Accessed 24/4/2010).

Nimbus 2.4 Documentation (Online). Available: http://www.nimbusproject.org/docs/2.4/index.html (Accessed 22/4/2010).

2009a. *Haizea* (Online). Available: http://haizea.cs.uchicago.edu/index.html (Accessed 24/4/10 2010).

2009b. Xen Cloud Platform 0.1 User Security. 1.0 ed.: Xen.org.

2009c. Xen Cloud Platform Administrator's Guide. *Release 0.1.* 1.0 ed.: Xen.org.

2009d. *Xen in the Cloud* (Online). Xen.org. Available: http://www.xen.org/files/Marketing/XeninCloud.pdf (Accessed 1/5/2010).

2010a. *abicloud Home* (Online). Available: http://community.abiquo.com/display/ABI15/Home (Accessed 2/5/2010).

2010b. *About openQRM* (Online). Available: http://www.openqrm.com/ (Accessed 5/5/2010).

2010c. *About Us* (Online). Available: http://www.abiquo.com/company/about-us.php (Accessed 7/5/2010).

2010d. *Architecture Overview* (Online). Available: http://abicloud.org/display/admin/Architecture+Overview (Accessed 2/5/2010).

2010e. *Open Source Home* (Online). Available: http://open.eucalyptus.com/ (Accessed 5/5/2010).

2010f. *OpenNebula 1.4 Architecture* (Online). Available: http://www.opennebula.org/documentation:rel1.4:architecture (Accessed 24/4/10 2010).

2010g. *OpenNebula Home* (Online). Available: http://www.opennebula.org/ (Accessed 23/4/10 2010).

2010h. *openQRM 4.6 release brings streamlined interface design and new professional cloud management tools* (Online). Available: http://www.openqrm-enterprise.com/about-openqrm/release-notes.html (Accessed 5/5/2010).

2010i. *Xen Cloud Platform - Advanced Virtualization Infrastructure for the Clouds* (Online). Available: http://xen.org/products/cloudxen.html (Accessed 1/5/2010).

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. & Zagorodnov, D. 2009. The Eucalyptus Open-Source Cloud-Computing System. *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid.* IEEE Computer Society.

Nurmi, D., Wolski, R., Obertelli, C. G. G., Soman, S., Youseff, L. & Zagorodnov, D. 2008. Eucalyptus : A Technical Report on an Elastic Utility Computing Archietcture Linking Your Programs to Useful

Systems. *UCSB Computer Science Technical Report Number 2008-10* (Online). Available: http://open.eucalyptus.com/documents/nurmi_et_al-eucalyptus_tech_report-august_2008.pdf.

Rechenburg, M. 2010. The OpenQRM Datacenter Management and Cloud Computing Platform. Germany: openQRM Enterprise GmbH.

Sotomayor, B., Montero, R. S., Llorente, I. M. & Foster, I. Year. Capacity Leasing in Cloud Systems using the OpenNebula Engine. *In:* Workshop on Cloud Computing and its Applications 2008 (CCA08), October 22-23, 2008 2008 Chicago, Illinois, USA.

Sotomayor, B., Montero, R. S., Llorente, I. M. & Foster, I. 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. *Internet Computing, IEEE,* 13, 14-22.