International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Infrastructure-as-a-Service and Comparative Study on Open-Source Cloud Infrastructure Solutions

R.P.Limbole[Å*] and G.P.Bhole[Å]

[Å]Dept of Computer Engineering & IT, VJTI,Mumbai,India

Accepted 12 July 2014, Available online 01 Aug 2014, **Vol.4, No.4 (Aug 2014)**

Abstract

Cloud computing is an attractive computing model since it allows for resources to be provisioned according on a demand basis, i.e., cloud users can rent resources as they become necessary. With IaaS, IT services can be delivered as a subscription service, eliminating up-front costs and driving down ongoing support costs. As with managed hosting, IaaS providers keep costs low by pooling resources and giving customers access to a shared facility. But a major difference is that IaaS resources are elastic and available on a self-service, on-demand basis. The convergence of those trends, with other advances of the last several years, has made it possible to take infrastructure outsourcing to a new level. Building on the foundation of managed service such as Infrastructure-as-a-Service (IaaS) has emerged as an easily deployed service that enables companies to flexibly and cost-effectively anticipate and evolve with their customers' rapidly changing business requirements. This paper analyses opens source solutions like XCP, Nimbus, OpenNebula, Eucalyptus, Enomaly, OpenStack for infrastructure in Cloud which uses KVM and Xen hypervisor.

Keywords: *Infrastructure-as-a-Service , Open-Source Solutions, KVM, Xen .*

1. Introduction

Cloud computing, at least conceptually, has been gaining rapid popularity as an easily accessible, strategic IT model. The downside of this widespread attention is that vendors have begun to provide their own definition for what comprises a cloud-based model, often based on the benefits that their specific service offering provides. To cut through the confusion, the U.S. National Institute for Standards and Technology (NIST) has come up with five essential characteristics that must be present for an offering to be considered ―cloud.

Scientific and business applications have an increasing demand for fast and scalable execution environments to deliver results for ever increasing problem sizes or concurrent requests in a requested time frame. Faced with the problem of costly maintenance and rapid deprecation following the Moore's law, companies and institutions operating scientific and business applications prefer to rent modern resource capabilities from specialized hosting companies instead of buying their own hardware. Cloud computing is the use of shared computing infrastructure to provide IT services in the form of a large pool of systems that are linked together. In essence cloud computing is a large group of computers working together to provide a service. Cloud computing has many different characteristics that define it. Some of the most prominent characteristics are:

a. On-demand self-service – Customers can unilaterally provision computing capabilities, such as server time and network storage, without requiring human interaction with the service provider. This is possibly the most important cloud computing criterion.

b. Broad network access - Capabilities and control of a cloud computing service must be available over the Internet (or other networks) using standard protocols, via a wide variety of platforms, including mobile devices.

c. Resource pooling - Physical and virtual resources are dynamically assigned and reassigned according to demand, resulting in cost savings to the customer.

d. Rapid elasticity - Near-immediate provisioning of capabilities, to quickly scale up, or down, according to demand.

e. Measured Service - Customers' use of the capabilities is monitored, controlled, reported, and charged with complete transparency, enabling a ―pay-as-you-consumer metering arrangement.

Advantages of IaaS versus other Services options

These are shown in table 1

2. Why IaaS?

While companies' reasons for considering IaaS differ, among SMBs and Enterprises alike, cost savings remains a key objective. A recent Yankee Group survey, focusing on cost savings, illustrates the top five motivations specified by respondents as reasons to use IaaS. Top Five Motivations to use Infrastructure-as-a- Service (IaaS):
Key Benefits of IaaS
1) Improved cash flow: With no up-front commitments and a pay-per-use consumption model, IaaS offers an

*Corresponding author: **R.P.Limbole**

Table 1 Comparing IT Sourcing Features

	IaaS	Managed Services	Conventional Data Centre
Provisioning	On-Demand, instant	On-demand, days to weeks	Weeks to months
Scaling	Instantly up or down	Weeks to months, up or down	Months, up only
Charges	No upfront, ”pay as you consume”	Upfront fees, minimum term commitment, fixed monthly fees	Capital-equipment purchase or lease, maintenance charges, plus bandwidth utilization fees
Self-service?	Yes, via desktop console	No	Not automated

improved start-up cash flow in comparison to conventional data center and IT service models.

2) Support of uncertain provision planning: The elasticity of IaaS enables customers to rapidly scale up resources as needed — even automatically — avoiding the undesirable consequences resulting from over- and under-provisioning.

3) Transparent metering and self-service management: The visibility and control IaaS offers over resources, activity and costs, represent a new level of IT management efficiency. Other benefits of IaaS include, for both SMBs and Enterprises, speed of deployment and increased functionality (such as easier backup and disaster recovery). For Enterprises specifically, IaaS brings particular advantages from both a tactical and strategic planning perspective.

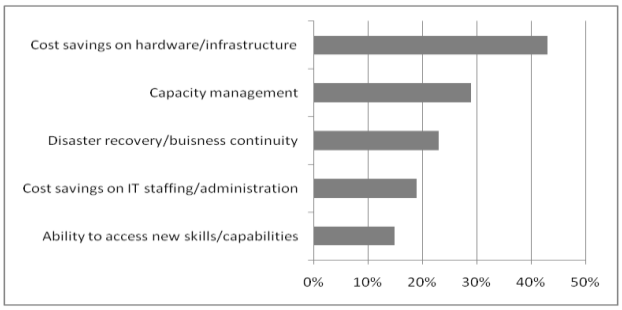

Fig.1 Advantages of IaaS

3. Challenges of Cloud Computing

The development of cloud computing solutions brings several technical challenges to cloud developers. These challenges can be grouped in three main areas: negotiation, decision, and operation. In the negotiation area, these are the challenges relative to how application developers interface with the cloud as well as the description of the cloud offerings. It includes also the definition of the programmability level that the cloud solution will offer. The decision area copes with the main problem that clouds faces behind the scenes: How virtual resources can be scheduled to meet user requirements, for example? Last, the operation area is associated with the enforcement of decisions and the communication between cloud elements. The following sub-sections discuss in details the main challenges in each one of these areas.

3.1. Negotiation

The negotiation area concerns itself with challenges relative to the interface between the application developers and the cloud. Generally, the interface between the cloud and application developers assumes the form of an Application Programming Interface (API), but, depending on the programmability level offered by the cloud, this API can be implemented in several ways ranging from a web-service based toolkit to control virtual machines in the cloud to a set of programming primitives used to develop distributed applications in the cloud. In addition to these basic functions, such APIs must allow developers to request – and control, possibly – additional functionalities offered by a cloud operator like service quality assessment, load balance, elastic application growth, backup strategies, and so on. There are still some other requirements that can be considered in the cloud API like geographical restrictions enforced to allocate virtual machines due to legal issues. One may think of some type of content or application that is strategically limited to a country or a region for copyright or security reasons.

At the present, APIs and the negotiation process offered by cloud providers follow a semi-automatic scheme where a human interacts with the cloud through programming primitives or a visual interface. But, next generation clouds can offer sophisticated ways to interact with human users through high-level abstractions and service-level policies. Such an interface type will need some formalism to specify both cloud offerings and application requirements as well as offering the support for an automatic negotiation process.

3.2 Decision

The main target of any cloud operator is to schedule developer applications aiming for the maximum utilization of cloud resources. A developer´s application covers, beyond the actual code, some additional information about application´s needs and services negotiated previously. In other words, one can abstract this additional information to some type of network virtualization demand with a topology formed by virtual nodes where the application runs and virtual links for communication. Thus, the cloud operator problem turns into that of selecting the best suitable physical resources to accommodate these virtual resources.

This careful mapping requires advanced strategies. The first challenge imposed by this optimization problem is that it is NP-hard(Andersen D.,2002) and hence any useful solution would need to relax some of its problem conditions and constraints to obtain an approximate solution in a polynomial time. The second challenge is to meet all the clauses negotiated with the developer. Depending on the nature of such contract, application scheduling algorithms will cope with quality restrictions, jurisdiction restrictions, elastic growth, and so on.

3.3 Operation

Metaphorically, one can say that while in the decision area the cloud operator must identify solutions for the ―brain of the cloud, in the operation area it must attack the problems of the ―limbs of the cloud, i.e., they must provide some form to enforce decisions. The enforcement here covers the communication protocols and the configuration of cloud elements.

A communication protocol can be used to monitor and reserve resources in the cloud. The cloud is composed by different elements like processing servers, switches, routers, links and storage components. Due to such heterogeneity, the communication between the decision-maker and elements puts a challenge on cloud design. Overall, existing cloud solutions use Web Services to provide communication with processing and storage nodes, but many communication elements do not support such implementations. Thus, cloud architects are using traditional traffic engineering protocols to provide reservation of network elements. One possible idea to cope with this challenge is to use smart communication nodes with an open interface to create new services in the node the emerging Openflow-enabled switches(N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. ,2008).

Node communication is just one part of the problem; the other one is to configure this. Here, the recent advances in server virtualization have provided several solutions for operators to benefit from.

4. Standardization efforts

A considerable challenge present in many of the raised discussions around the cloud is related to the need for standardization. All the three areas presented in Section 3 face the standardization challenge in some way, but the main challenge occurs in the negotiation area. Currently, cloud providers offer proprietary interfaces to access their services. This locks users within a given provider as they cannot migrate their applications and services easily between cloud providers(Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.,2009). It is hoped that cloud providers see such a problem and work together to offer a standardized API based on open standards like SOAP and REST.

An important effort in the standardization comes from the Open Cloud Manifesto(OpenCloud.,2009). This is an initiative supported by hundreds of companies that aims to discuss with cloud organizations a way to produce open standards for cloud computing. Their major doctrines are collaboration and coordination of efforts on the standardization, adoption of open standards wherever appropriate, and the development of standards based on customer requirements. Participants of the Open Cloud Manifesto through the Cloud Computing Use Case group produced an interesting white paper(OpenCloud.,2010) highlighting the requirements that need to be standardized in a cloud environment to ensure interoperability in the most typical scenarios of interaction – Use Cases – in cloud computing.

5. Open-source solutions for Cloud Computing

Due to the large growth of cloud computing, there are several solutions in this area. This article is focused on open source solutions, highlighting their main characteristics and architectures proposed.

5.1. Xen Cloud Platform (XCP)

The Xen hypervisor(Citrix Systems,2010) is a solution for infrastructure virtualization that provides an abstraction layer between servers' hardware and the operating system. A Xen hypervisor allows each physical server to run several ―virtual servers handling the operating system and its applications from the underlying physical server. The Xen solution is used by many cloud solutions such as Amazon EC2, Nimbus and Eucalyptus.

Recently, Xen.org announced the Xen Cloud Platform (XCP)(Citrix Systems,2010) as a solution for cloud infrastructure virtualization. But, differently from existent open source cloud solutions, XCP does not provide the overall architecture for cloud services. Their goal is to provide a tool to cope with automatic configuration and maintenance of cloud platforms.

XCP Resource Pool

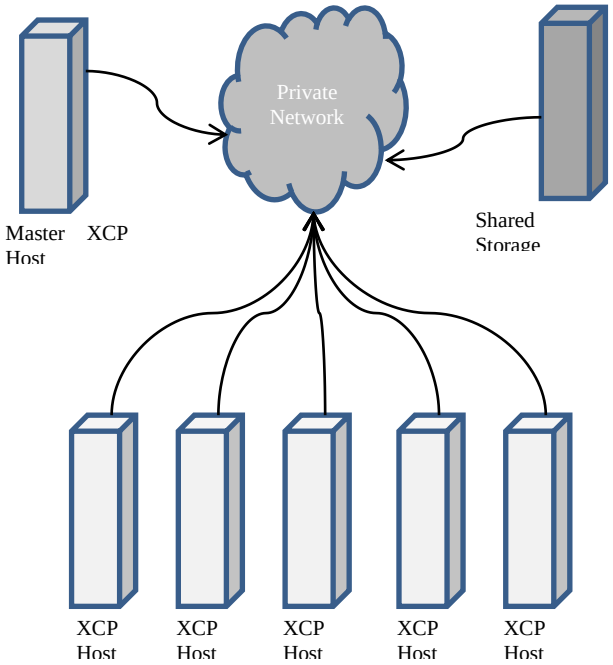

Fig.1 XCP Architecture

The XCP architecture (Figure 2) is based on the XCP hosts that are responsible to host the virtual machines. According to(Xen.org, 2009), these hosts are aggregated in a XCP resource pool and using a Shared Storage the virtual machines can be started and restarted on any XCP host. The Master XCP host offers an administration interface and forwards command messages to others XCP hosts.

5.2. Nimbus

Nimbus(Keahey, K, 2009) is an open source solution (licensed under the terms of the Apache License) to turn clusters into an Infrastructure as a Service (IaaS) for Cloud Computing focusing mainly on scientific applications.
This solution gives to users the possibility to allocate and configure remote resources by deploying VMs – known as Virtual Workspace Service (VWS). A VWS is a VM manager that different frontends can invoke. To deploy applications, Nimbus offers a ―cloud kit configuration that consists of a manager service hosting and an image repository. The workspace components are shown in Figure 3.

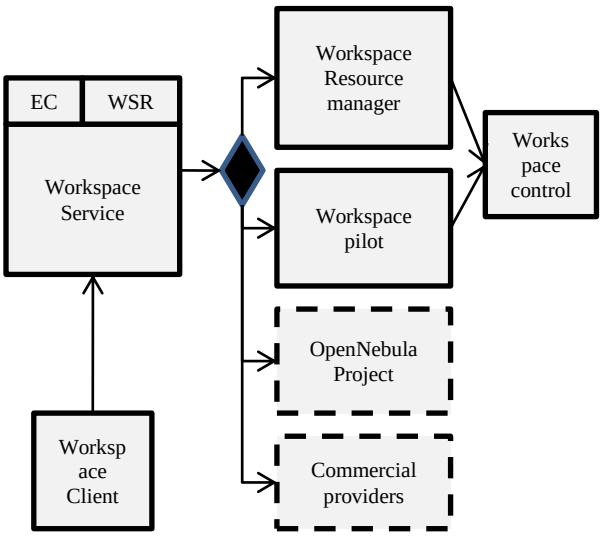

Fig.3 Nimbus workspace Components

1) Workspace service: is web services based and provides security with the GSI authentication and authorization. Currently, Nimbus supports two frontends: Amazon EC2 and WSRF.
2) Workspace control: is responsible for controlling VM instances, managing and reconstructing images, integrating a VM to the network and assigning IP and MAC addresses. The workspace control tools operate with the Xen hypervisor and can also operate with KVM1.
3) Workspace resource management: is an open source solution to manage different VMs, but can be replaced by other technologies such as OpenNebula.
4) Workspace pilot: is responsible for providing virtualization with few changes in cluster operation. This component handles signals and has administration tools.

5.3. OpenNebula

OpenNebula(OpenNebula Project, 2010) is an open source toolkit used to build private, public and hybrid clouds. It has been designed to be integrated with networking and storage solutions and to fit into existing data centers. The OpenNebula architecture (Figure 4) is based on three basic technologies to enable the provision of services on a distributed infrastructure: virtualization, storage and network. All resource allocation is done based on policies.

The Cumulus Project(Wang, L., Tao, J., Kunze, M., Rattu, D. and Castellanos, A, 2008) is an academic proposal based on OpenNebula. Cumulus intends to provide virtual machines, virtual applications and virtual computing platforms for scientific applications. Visualizing the integration of already existing technologies, the Cumulus project uses HP and IBM blade serves running Linux and Xen hypervisor.

The Cumulus networking solution was called the ―*forward* mode, where users do not need to specify any network configuration information. Instead the backend servers are responsible for allocating a dynamic IP address for a VM and returning these to the users, making such networking solution transparent to the users. The Cumulus design is a layered architecture (Figure 5) with three main entities: Cumulus frontend, OpenNebula frontend, and OS Farm. This proposal focuses on reaching scalability and autonomy of data centers.

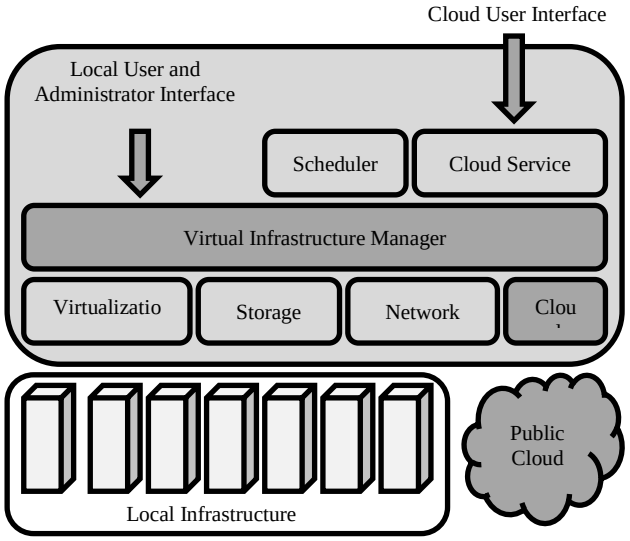

Fig.4 OpenNebula Architecture

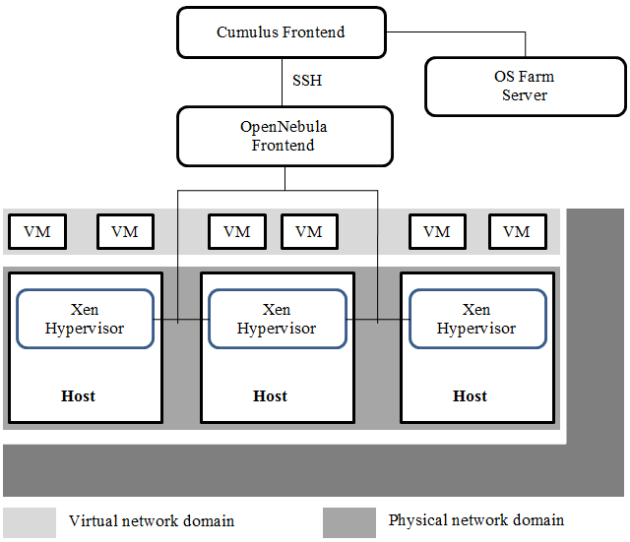

Fig.5 Cumulus Architecture

1) Cumulus frontend: the Cumulus frontend is the access point for a Cumulus system and is responsible for handling VM requirements.

2) OpenNebula frontend: the OpenNebula frontend provides an interface to manage the distributed blade servers and the resources for VM deployment. To administrate a common user system, Cumulus uses NIS (Network Information System) and NFS (Network File System) to manage shared directory. Moreover, OpenNebula was merged with secure infrastructure solutions, such as LDAP (Lightweight Directory Access Protocol) and the Oracle Cluster File System.
3) OS Farm: the OS Farm is a tool for VM template management that operates to generate and to store Xen VM images and virtual appliances.

5.4. Eucalyptus

Eucalyptus(Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. And Zagorodnov, D,2009) is an open source cloud computing framework focused on academic research. It provides resources for experimental instrumentation and study. Eucalyptus users are able to start, control, access and terminate entire virtual machines. In its current version, Eucalyptus supports VMs that run atop the Xen supervisor(Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A, 2003).

According to, the Eucalyptus project presents four characteristics that differentiate it from others cloud computing solutions: a) Eucalyptus was designed to be simple without requiring dedicated resources; b) Eucalyptus was designed to encourage third-party extensions through modular software framework and language-agnostic communication mechanisms; c) Eucalyptus external interface is based on the Amazon API (Amazon EC2) and d) Eucalyptus provides a virtual network overlay that both isolates network traffic of different users and allows clusters to appear to be part of the same local network. The Eucalyptus architecture is hierarchical (Figure 6) and made up of four high level components, where each one is implemented as a stand-alone web service.
1) Node Controller (NC): this component runs on every node that is destined for hosting VM instances. An NC is responsible to query and control the system software (operating system and hypervisor) and for conforming requests from its respective Cluster Controller. The role of NC queries is to collect essential information, such as the node's physical resources (e.g. the number of cores and the available disk space) and the state of VM instances on the nodes. NC sends this information to its Cluster Controller (CC). NC is also responsible for assisting CC to control VM instances on a node, verifying the authorization, confirming resources availability and executing the request with the hypervisor.
2) Cluster Controller (CC): this component generally executes on a cluster frontend machine, or any machine that has network connectivity to two nodes: one running NCs and another running the Cloud Controller (CLC). A CC is responsible to collect/report information about and schedule VM execution on specific NCs and to manage virtual instance network overlay.
3) Storage Controller (Walrus): this component is a data storage service that provides a mechanism for storing and accessing virtual machine images and user data. Walrus is based on web services technologies and compatible with Amazon's Simple Storage Service (S3) interface(Amazon, 2006).
4) Cloud Controller (CLC): this component is the entry-point into the cloud for users. Its main goal is to offer and manage the Eucalyptus underlying virtualized resources. CLC is responsible for querying node managers for resources' information, making scheduling decisions, and implementing them by requests to CC. This component is composed by a set of web services which can be grouped into three categories, according their roles: resource services, data services, and interface services.

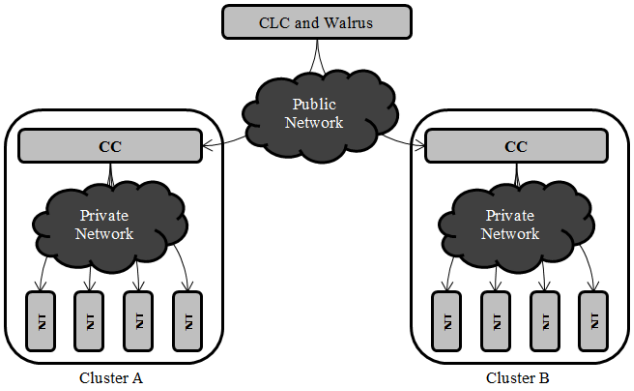

Fig.6 Eucalyptus Architecture

Ubuntu Enterprise Cloud (UEC)2 is an Amazon EC2 like infrastructure and is powered by Eucalyptus. Its main goal is to provide a simple process of building and managing internal infrastructure for cloud. The Ubuntu 9.04 Server Edition is integrated with Eucalyptus that uses the KVM hypervisor. The UEC architecture is based on the Eucalyptus architecture in which each elements is an independent web service that publishes a Web Service Description Language (WSDL) document defining the API to interact with it.

Furthermore, UEC defines three layers for security: authentication and authorization, network isolation and Machine Instance Isolation (MInst). The authentication and authorization layer is responsible for locally generated X.509 certificates; the network isolation layer is important to prevent eavesdropping of network traffic and; the MInst layer consists of Networking isolation, Operating System isolation, and Hypervisor based machine isolation.

5.5 Enomaly Elastic Computing Platform

Enomaly ECP Community Edition under the AGPL license is the open source cloud solution offered by Enomaly Inc. This version focuses on virtual machine administration in small clouds environments. Compared with the Enomaly commercial solution (called Service Provider Edition), the Enomaly open source edition suffers from many restrictions, such as limited scalability, no capacity control mechanism, no support for accounting and metering, and so on(Enomaly, 2009).

5.6 OpenStack

OpenStack Compute (Nova)(OpenStack Compute) is a cloud computing fabric controller, which is the main part of an IaaS system. It is designed to manage and automate pools of computer resources and can work with widely available virtualization technologies, as well as bare metal and high performance computing configurations. KVM and Xen are available choices for hypervisor technology, together with Hyper-V and Linux container technology such as LXC.

It is written in Python and uses many external libraries such as Eventlet (for concurrent programming), Kombu (for AMQP communication), and SQLAlchemy (for database access). Compute's architecture is designed to scale horizontally on standard hardware with no proprietary hardware or software requirements and provide the ability to integrate with legacy systems and third-party technologies.

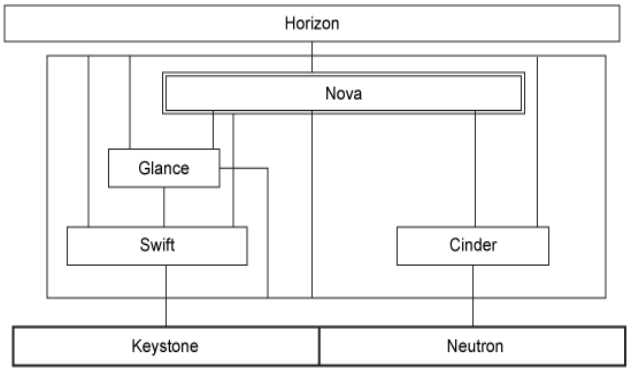

Fig.7 OpenStack architecture

Three elements interact with all the components in the system. Horizon is the graphical UI that administrators can most easily use to manage all the projects. Keystone handles the management of authorized users, and Neutron defines the networks that provide connectivity between the components.

Nova can arguably be considered the core OpenStack. It handles the orchestration of workloads. Its compute instances usually require some form of persistent storage which can be either block-based (Cinder) or object-based (Swift). Nova also requires an image to launch an instance. Glance handles this request, whereby it can optionally use Swift as its storage back end.

The OpenStack architecture had endeavored to make each project as independent as possible which gives users the option to deploy only a subset of the functionality and integrate it with other systems and technologies that offer similar or complementary functions. Nonetheless, this independence shouldn't mask the fact that a fully functional private cloud is likely to require virtually all the functionality to operate smoothly, and the elements will need to be tightly integrated.

6. Discussion

As pointed out earlier in this paper, there are several solutions for cloud computing. Each solution presents a different vision about cloud architecture and implementation. Moreover, each approach has an implication that directly impacts its business model: the closer to the hardware level, the more options a user can handle but at the cost of having to configure her cloud (more configuration flexibility). Amazon EC2 and IBM Capacity on Demand (CoD) are solutions that offer to their users this configuration flexibility. In this business model, users can choose and configure computational resources at the hardware level and OS levels. At the other extreme, solutions like Google App Engine and Windows Azure, try to turn development easy to their users, but at the same time, confine them to specific APIs and software platforms. Moreover, solutions like JoliCloud4 are more limited as they offer a single service (operating system). In the middle, there are solutions that offer a middleware-like approach to users, where the hardware resources can be configured and handled subject to some restrictions and where applications can also be developed. All the presented open-source solutions and the cited commercial solutions are categorized into Figure 8. The graphic compares solutions and their business model (hardware, middleware and user level) according to configuration flexibility.

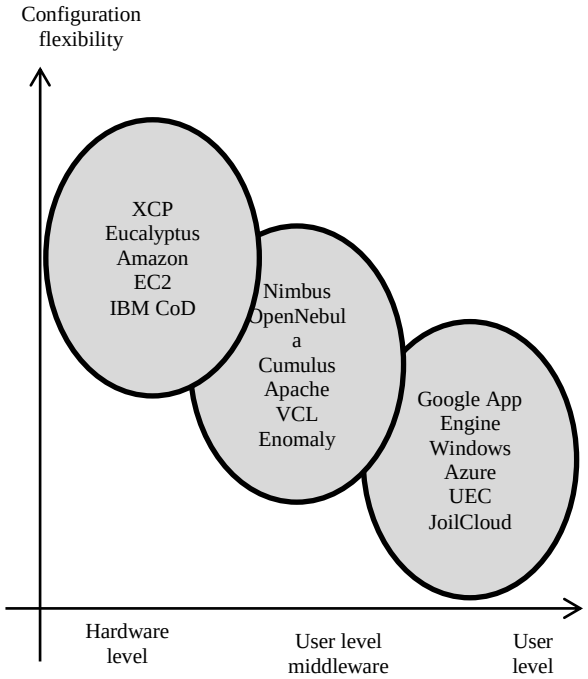

Fig.8 Cloud Computing Solutions

Finally, Table 2 presents a comparative board of the open source cloud solutions described in this paper, in terms of the service type (IaaS, PaaS, and SaaS), the main characteristics, and the infrastructure technologies. The table also cites some users of each cloud solution.

Conclusion

The promise of cloud computing has long been a new height of convenience—easily and rapidly provisioned pay-per-use computing resources, scaling automatically and instantly to meet changing demands. Emerging at the convergence of major computing trends such as

Table 2 Comparison between open-source cloud computing solutions for IaaS

Solutions	Service	Main Characteristics	Infrastructure	Used by
XCP	IaaS	Only a tool for automatic maintenance of clouds	Xen	XCP community
Nimbus	IaaS	Aims to turn legacy clusters into IaaS Clouds	Xen hypervisor and KVM	Brookhaven National Labs
OpenNebula	IaaS	Policy-driven resource allocation	Xen hypervisor	Cumulus Project
Eucalyptus	IaaS	Hierarchical Architecture	Xen hypervisor and KVM	UEC
Enomaly	IaaS	Open version is focused in small clouds	Xen, KVM and VirtualBox	Several Companies
OpenStack	IaaS	Manage and Automate pools of resources and work with widely available virtualization technologies	KVM and Xen are available choices for hypervisor technology	NASA

virtualization, service oriented architectures, and standardization of the Internet, IaaS comes closer than ever before to fulfilling that vision. As with disruptive business models from the past, certain technical, legal, and personnel challenges must be overcome before IaaS will enter the mainstream. Nonetheless, organizations would do well to begin the evaluation process by:

1) Amassing available literature on IaaS
2) Contacting IaaS providers for a consultation and audit of current practices
3) Developing an accurate TCO of current IT solutions
4) Working with an IaaS provider to develop a migration plan
5) Testing IaaS with a new application launch or nonbusiness-critical application
6) Benchmarking costs and performance of current solutions vs. IaaS candidate applications

Companies that effectively leverage the benefits of an IaaS environment may be able to gain an edge in a rapidly evolving economy. There is a clear need for the standardization of current cloud platforms at least of terms of interface, negotiation and access through Web services. Understandably, this is a considerable task as many clouds use different abstraction levels, some are generic whereas others focus on a specific application domain, etc. Some initial steps have been taken into this direction with the setup of the Open Cloud Manifesto, an initiative supported by hundreds of companies. In the mean time we will continue to see some clouds such as Nimbus implementing a number of front ends (e.g. Amazon EC2 and WSRF) to ensure access to their existing users. Interestingly, some solutions such as the OpenNebula have been first to adopt policies for resource management. The use of policies remains a challenge in many areas and clouds may benefit from it. It is also important to acknowledge the leadership and string presence of academic efforts such as Eucalyptus and Xen. These have been at the forefront of innovation supported by the many commercial cloud systems that currently are based on these. These efforts are expected to continue as more work is needed to remove much of the mysticism and conflicts surrounding the use of clouds.

References

Andersen, D. (2002), Theoretical Approaches to Node Assignment, *Unpublished manuscript. http://www.cs.cmu.edu /dga/papers/ Andersen -assign.ps*

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. (2008), OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I. (2009), Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, Elsevier B. V.*

OpenCloud. (2009), The Open Could Manifesto. *http://www.opencloudmanifesto.org/.*

OpenCloud. (2010), Cloud Computing Use Cases, *Whitepaper - Version 3.0. http://groups.google.com/group/cloud-computing-use cases?hl=en.*

Citrix Systems. (2010), Xen Hypervisor, *http://xen.org/products/ xenhyp.html.*

Citrix Systems. (2010),Xen Cloud Platform - Advanced Virtualization Infrastructure for the Clouds, *http://www.xen.org/products/cloudxen.html*

Citrix Systems. (2010), The Xen Cloud Project,The Citrix Blog. Available at *http:// community.citrix .com/pages /viewpage.action?pageId =81690805*

Xen.org. (2009) "Xen Cloud Platform Administrator's Guide: Release 0.1".

Keahey, K. (2009), Nimbus: Open Source Infrastructure-as-a-Service Cloud Computing Software, Workshop on adapting applications and computing services to multi-core and virtualization, CERN, Switzerland.

OpenNebula Project. (2010), OpenNebula.org – The Open Source Toolkit for Cloud Computing, *http://www.opennebula. org/.*

Wang, L., Tao, J., Kunze, M., Rattu, D. and Castellanos, A. (2008), The Cumulus Project: Build a Scientific Cloud for a Data Center. *In: Workshop on Cloud Computing and Its Applications, Chicago, USA.*

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. And Zagorodnov, D. (2009), The Eucalyptus Open-Source Cloud Computing System. *In: 9th IEEE/ACM International Symposium on Cluster Computing and the Grid.*

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A. (2003), Xen and the art of virtualization. *In: 19th ACM symposium on Operating systems principles, New York, NY, USA.*

Amazon. (2006), Amazon simple storage service API Reference,*http://docs.amazonwebservices.com/AmazonS3/200 6-03-01/.*

Enomaly. (2009), Enomaly – Elastic Computing. *http://src. enomaly.com/.*

OpenStack Compute: An Overview http:// http:// www.openstack.org/downloads/ openstack- compute-datasheet.pdf