A Comparative Study of Current Open-source Infrastructure as a Service Frameworks

Theo Lynn, Graham Hunt, David Corcoran, John Morrison and Philip Healy
Irish Centre for Cloud Computing, Dublin 9, Ireland

Keywords: Cloud Computing, Open Source, IaaS, Openstack, Cloudstack, Opennebula, Eucalyptus.

Abstract: With the growth of cloud computing in recent years, several commercial and open source IaaS frameworks have emerged. The development of open source IaaS solutions offers a free and flexible alternative to commercial cloud services. The main contribution of this paper is to provide a qualitative comparative of current open-source IaaS frameworks. Existing research papers examining open source IaaS frameworks have focused on comparing OpenStack with a small number of alternatives. However, current research fails to adequately compare all major open source frameworks in a single study and notably lacks the inclusion of CloudStack. Our research paper provides the first overview of the five main open source cloud IaaS frameworks – OpenStack, CloudStack, OpenNebula, Eucalyptus and Nimbus. As such, this review provides researchers and potential users with an up to date and comprehensive overview of the features of each solution and allows for an easy comparison between the open source solutions.

1 INTRODUCTION

Cloud Computing technologies have seen significant adoption in recent years by enterprises, researchers and individuals. It is forecast that the Cloud Computing industry will reach a market size of $241 billion by 2020 (Reid and Kilster, 2011). Cloud Computing has become an important tool in delivering Infrastructure as a Service (IaaS) for users that require a high level of flexibility and management of their software stack.

The last eight years has seen the rapid proliferation of vendors in the IaaS market, which includes major enterprise players such as Amazon (Amazon, n.a.), Microsoft (Mircosoft, n.a), Google (Google, n.a.), and IBM (Google, n.a.). While these services may offer flexibility to the user, commentators cite cost, lack of portability and lack of interoperability as drawbacks to enterprise IaaS (Mahjoub et al., 2011; Zhang et al., 2013). With each vendor promoting its own infrastructure and incompatible standards, users can suffer from vendor lock-in (Mahjoub et al, 2011; Zhang et al., 2013; Wen et al., 2012).

In some environments it may be preferable for an organization to develop their own cloud, based on an open source framework. Open source frameworks offer an alternative to enterprise clouds by offering the freedom to modify the source code and build a cloud that is pluggable and open to extensions while reducing costs and avoiding vendor lock-in (Zhang et al., 2013; Wen et al. 2012; Bist et al., 2013). The development of open source solutions is of particular importance for the further proliferation of private and hybrid clouds (Sefraoui, 2012).

The decision to select the most appropriate open source solution can be a challenge for organizations given each framework's specific characteristics. While previous studies (Zhang et al., 2013; Dukarić, R. and Jurič, 2013; Endo et al., 2010; Yaday, 2013) have compared these frameworks, most have focused only on OpenStack and OpenNebula, neglecting CloudStack or have focused on comparing frameworks with different hypervisors (Ristov et al., 2013; Cordeiro , 2010).

The purpose of this paper is to provide a qualitative review of the top five open source IaaS frameworks and present a comparative analysis to aid in framework selection decisions. The frameworks included in this paper have been chosen based on literature reviews and perceived industry acceptance. The open source frameworks detailed in this paper are OpenStack, CloudStack, OpenNebula, Eucalyptus and Nimbus.

The paper is structured as follows. In Section II, we present a background to the study providing an

Lynn T., Hunt G., Corcoran D., Morrison J. and Healy P..
A Comparative Study of Current Open-source Infrastructure as a Service Frameworks.
DOI: 10.5220/0005423300950104
In *Proceedings of the 5th International Conference on Cloud Computing and Services Science* (CLOSER-2015), pages 95-104
ISBN: 978-989-758-104-5

overview of the benefits of Cloud Computing adoption and discuss the different service models. Section III provides a qualitative review concerning the five chosen frameworks and examines their components. This section also provides a short overview of closed vs. open source IaaS solutions.

Drawing on the review presented in Section III, Section IV provides a comparative analysis of the identified frameworks based on selected criteria. Finally, Section V will discuss our conclusions and recommendations for future developments.

2 BACKGROUND

The National Institute of Standards and Technology (NIST) defines Cloud Computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance, 2011). Today it is the most widely cited and accepted definition. This definition can be expanded to include five additional characteristics of cloud computing. These characteristics have also become widely accepted and cited; they are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The NIST definition of cloud computing outlines three service models: software as a service (SaaS); platform as a service (PaaS); and infrastructure as a service (IaaS). In addition, it outlines four deployment models: private cloud; community cloud; public cloud; and hybrid cloud.

2.1 Service Models

Cloud Computing is typically classified according to a variety of service models. In recent years, service models such as Storage as a Service (Fielder et al., 2012) and Business Process as a Service (Lynn et al., 2014) have emerged. However, the most common differentiation of service models remains as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), all of which have been given a definition by NIST.

Software as a Service is when the supplier provides working applications running on their own cloud infrastructure, typically these applications would be accessed through a web browser or some other type of web interface or application (Mell and Grance, 2011).

Platform as a Service is when the supplier provides the user with the tools required for deploying an app onto the cloud infrastructure (Mell and Grance, 2011).

Finally, Infrastructure as a Service, the focus of this paper, is when the supplier provides the consumer with processing, memory, storage, networks, and other computing resources (Mell and Grance, 2011).

3 OPEN SOURCE IaaS FRAMEWORKS

Enterprise Solutions (or closed frameworks) typically supply servers and storage to their customers using their own infrastructure through a public cloud development (Zhang et al., 2013). Open source IaaS frameworks take advantage of open source code which can be modified by users to create a single functional package that can be applied to a network of servers and storage to produce IaaS (Zhang et al., 2013). While open source solutions are typically used to develop private clouds they are also suitable for hybrid and public cloud development models (Salih and Zang, 2012; Ristov et al., 2013; Wen et al., 2012). Recent acquisitions such as Citrix's purchase of CloudStack (Keeps, 2014) suggest that we are likely to see further development of public clouds based on open source frameworks, leading to improved interoperability between different clouds.

The main player offering enterprise solutions is Amazon Web Services, with growing competition from competitors including Microsoft, Google, and IBM (Gartner, 2014). The remainder of this paper will discuss open source frameworks.

Five major projects, OpenStack, Apache CloudStack, OpenNebula, Eucalyptus, and Nimbus dominate the market for Open Source IaaS. A survey conducted in July 2014, by Linux.com, of 550 Open Source cloud computing experts found that OpenStack, CloudStack, OpenNebula, and Eucalyptus are the most popular Open Source IaaS options with all others receiving less than 2% of the result (Williams, 2014). OpenStack was favoured with 63 per cent of the vote, followed by CloudStack, OpenNebula, and Eucalyptus, respectively. OpenStack was also found to be the most popular overall Open Source Cloud Computing project. One of the reasons suggested for OpenStack's popularity is that the strength of the OpenStack community and the prestige and size of the companies who back OpenStack has given it a significant advantage over its alternatives (Voras et al., 2013). Another reason suggested is that with the maturation of the market there has been a focus on the bigger projects available in search of a de facto standard (Voras et al., 2013).

An alternative view is that no one Open Source IaaS framework will dominate the market, as each framework is suited to different end use scenarios or applications. For example it is proposed that OpenNebula is more suited to security-sensitive and smart-city scenarios while OpenStack is more suited for time-constrained scenarios (Kostantos, 2013).

At this stage in the market's maturation, the likelihood of new entrants is slim (Voras et al., 2013). Considering the lead the five mentioned projects have over their alternatives, we have decided to focus our analysis on these projects for the purpose of this paper. In the remainder of this section we provide a short discussion of the qualitative features of the different open source IaaS frameworks.

3.1 OpenStack

Announced in 2010, OpenStack is an entirely open sourced IaaS project for private and public clouds. Initially developed by Rackspace Hosting and NASA, it has since developed into a global collaboration of developers and cloud computing technologists. With over 4500 members, 850 companies, and support from major tech industry companies, OpenStack has a very powerful foundation. The OpenStack Open Source Cloud Mission is "to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable" (Openstack, n.a.).

OpenStack has three core official software programs, Compute (Nova), Object Storage (Swift),

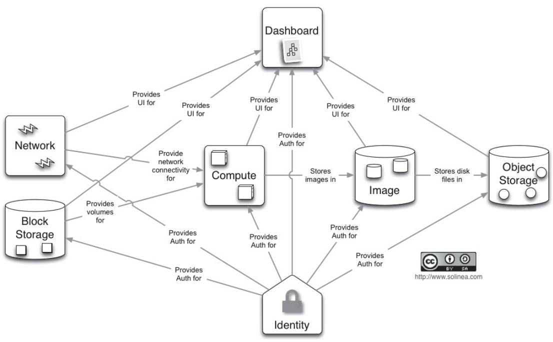

Figure 1: Conceptual OpenStack Architecture, Source: Openstack, n.a..

and Image Service (Glance).

Nova is the computing engine behind OpenStack. It is the cloud computing fabric controller and the central element of the IaaS system (Openstack, n.a.). It manages all the compute resources of the OpenStack cloud. It includes an API (nova-api) server that accepts requests for creating virtual machines (VM), and their local or remote disks (Baset, 2012), and associated metadata in a VM. Other processes within Nova include:

- nova-schedule – takes a VM request and determines where it should be placed.
- nova-compute – creates and terminates VM instances through a hypervisors' API. It includes support for Xen, XenServer/XCP, KVM, VMware vSphere, LXC, QEMU, UML, Hyer-V, etc.

Swift offers cloud storage software enabling the storage and access of objects across different nodes using a simple API. It is designed to be extremely scalable in size and capacity.

Glance provides image services for OpenStack. It provides a catalogue and repository of VM images, also allowing them to be used as templates when deploying new VM instances (Kumar et al., 2014).

Other official OpenStack programs include Cinder, Neutron, Horizon, Keystone, Ceilometer, Heat, and Trove.

Cinder is the block storage component of OpenStack. It provides persistent block level storage devices for use with OpenStack compute instances.

Neutron is OpenStack Networking; it provides networking capabilities to OpenStack and manages networks and IP addresses (Kumar et al., 2014).

Horizon is the dashboard behind OpenStack. It provides a graphical interface for users and administrators.

Keystone provides identity services to OpenStack. It is a means of access to the OpenStack services available to different users.

Ceilometer provides billing services to individual users of the cloud. It also keeps an account of the users' usage of different components of OpenStack.

Heat uses a template based orchestration mechanism to launch multiple composite cloud applications. The templates are in the form of text files that are treated like code and define what resources are necessary for the code.

Trove is a database as a service for OpenStack. It provides functionality for both relational and non-relational database engines.

The latest OpenStack release, Icehouse, was released in April 2014. This release saw approximately 350 new features and 2,902 bug fixes added. Features added included support for rolling upgrades, the new feature discoverability that improves workflow, mandatory testing for external drivers, automated scaling of additional resources across the platform.

3.2 CloudStack

CloudStack was developed by Cloud.com and in 2010 released the majority of the project's source code. In 2011 CloudStack were acquired by Citrix who subsequently released the remainder of CloudStack's code. In April 2012 Citrix donated CloudStack to the Apache Software Foundation (ASF). It was accepted into the Apache Incubator and then graduated from the Apache Incubator in March 2013 to become a Top Level Project of ASF (Cloudstack, n.a.) CloudStack doesn't have quite the same foundation of users as OpenStack but a number of large companies do use Apache CloudStack including Apple, British Telecom, Dell, Nokia, and Fujitsu.

CloudStack consists of the management server and the resources to be managed. The management server is the CloudStack software that manages the cloud resources and allocates resources in the cloud deployment. It provides a web interface for administrators and end users, as well as API interfaces for CloudStack API and also the Amazon EC2 interface. The management server also manages assignment of guest VMs, assignment of IP addresses, allocates storage during VM instantiation, and manages snapshots, disk images, and ISO images. It is a single point of configuration for the IaaS cloud and during deployment the user informs it of the resources to be managed.

The CloudStack deployment is organised with hosts contained within clusters, clusters contained within pods, pods contained within zones, and zone contained within regions. Regions are the largest available units in the CloudStack deployment.

Regions essentially consist of a number of available zones, with zones roughly being the equivalent of a data enter. Regions can be used to provide fault tolerance and achieve higher availability and scalability. Zones, as mentioned typically correspond to a single data centre, but a data centre can also contain multiple zones. Pods often represent a single rack. Clusters are a way to group hosts; hosts in a cluster will all have identical hardware and run the same hypervisor. Hosts are single computers and provide the resources that run guest VMs. Figure 2 provides an outline of the

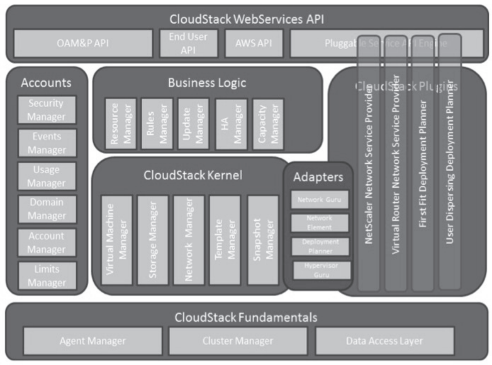

Figure 2: CloudStack Architecture, Source: Cloudstack.

conceptual architecture of a simple CloudStack deployment, note that does not include regions since these would not be necessary for a simple smaller scale deployment (Cloudstack).

The latest version of Apache CloudStack is 4.4.0 and was released in August 2014 and had a number of updates including updating to the latest version of Java.

3.3 OpenNebula

OpenNebula was initially released in 2008 and now operates as an open source project. While it is most commonly used for private cloud, it also supports public and hybrid clouds. The OpenNebula mission is "to become the simplest cloud enabling platform for the enterprise" and their purpose is "to bring simplicity to the private and hybrid enterprise cloud". OpenNebula has a large user base with telecommunication companies (e.g. Telefonica), system integrators (e.g. Logica), supercomputing centers (e.g. SARA), etc. (Kostantos et al., 2013). Additionally, in March 2010 the main authors of OpenNebula founded C12 Labs to provide a value-added professional service, which many enterprises require for internal adoption, and to enhance OpenNebulas long-term sustainability (Yadav, 2013).

The OpenNebula architecture is a classical cluster with a front end and a set of cluster nodes to run the VMs. At least one physical network is needed for connecting all cluster nodes with the front end (OpenNebula, n.a.). This architecture is flexible and modular, making it easier to integrate multiple storages, network infrastructures and hypervisor technologies than using OpenStack (Wen, 2012).

The OpenNebula software consists of three layers: tools, drivers, and core. An example of an

Figure 3: OpenNebula Architecture, Source (OpenNebula, n.a.).

architecture reference model for IaaS clouds is shown in Figure 3.

The **Tools** layer provides interfaces to communicate with users and allows users to manage VMs through the interfaces, which include command line interface (CLI) and libvirt API (Wen, 2012). This layer also contains a scheduler that manages the functionality of the core layer (Wen, 2012).

The **Drivers** layer contains components that communicate directly with the underlying operating system and capture the underlying infrastructure as an abstract service (Wen, 2012).

The **Core** layer (Figure 3) contains the components used to perform user requests and control resources. In this layer, disk images for VMs are stored using datastores (referred to as Image Repositories in previous releases). The monitoring subsystem gathers information on the hosts and the VMs, such as basic performance indicators and capacity consumption, and it also contains the authentication system, which comes with a built-in user/password authentication driver and the choice from SSH Authentication, X509 Authentication, and LDAP Authentication also.

The latest OpenNebula update is version 4.8.0, Lemon Slice, released on the 12th of August 2014. The release saw improvements to the hybrid model, changes to virtual networks that can now include any combination of ranges to accommodate any address distribution, and several other improvements throughout every other OpenNebula component.

3.4 Eucalyptus

Eucalyptus was developed by the University of California-Santa Barbara as an open source Infrastructure as a service and was released in 2008. It allows for the installation of private and hybrid clouds. Eucalyptus target is hybrid installations with its compatibility with Amazon Web Service AWS. Eucalyptus' scalability is limited in comparison to some of the alternative open source IaaS solutions and, as such, has lost some of the popularity it had in the earlier days of its release (Sefraoui et al., 2012). The introduction of Scalable Object Storage has helped improve the scalability of Eucalyptus.

Eucalyptus has six components (one being an optional component) (Ristov et al., 2013): Cloud Controller (CLC), Scalable Object Storage (SOS), Cluster Controller (CC), Storage Controller (SC), VMware Broker components and Node Controller (NC). These components are divided into three layers: Cloud Layer, Cluster Layer and Cloud Controller.

Cloud Controller is contained in the Cloud Layer and is the entry-point for administrators, developers, project managers, and end users. It queries the other components in Eucalyptus for information and makes requests to the Cluster Controllers. It is responsible for exposing and managing the underlying virtualized resources and offers a web-based administrative interface.

Scalable Object Storage is also contained in the Cloud Layer; this is the Eucalyptus equivalent to Amazon Web Services Simple Storage Service (S3). SOS allows for the implementation of scale-out storage that implements the S3 interface. Eucalyptus also provides a basic storage solution known as Walrus, suitable for smaller cloud deployments.

Cluster Controller is contained in the Cluster Layer. It communicates with the SC and NC and manages the execution of VM instances on specific NCs. The CC must have network connectivity to the machines running the CLC and the machines running the NC.

Storage Controller is also contained in the Cluster Layer. It communicates with the CC and the NC and manages the VM volumes and snapshots from volumes in the cloud. This component functions similarly to the Amazon Elastic Block Store.

The **VMware Broker** is the final component in the Cluster Layer and is the only optional component. It mediates interactions between the CC and VMware and can connect directly to ESX/ESXi hosts or to vCenter Server.

Node Controller is the sole component in the Node Layer. The Node Controller is written in C, hosts the VM instances and manages the virtual network endpoint (Eucalyptus, n.a.).

The latest release of Eucalyptus is version 4.0.1, a maintenance update for version 4.0.0 that was released on the 30th of May 2014. Some of the

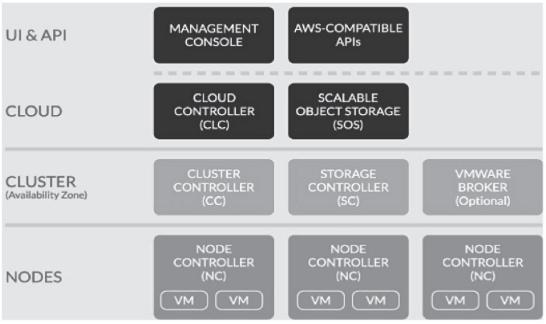

Figure 4: Eucalyptus Architecture, Source: Eucalyptus, n.a..

changes in this release include the possibility to have multiple user-facing services, the possibility to use different object storage backends, and more improvements throughout most of the Eucalyptus software.

3.5 Nimbus

Nimbus is an open source IaaS project that was initially released in 2009. Nimbus' mission is "to evolve the infrastructure with emphasis on the needs of science". Nimbus is built with three goals in mind:

- Enable resource owners to provide their resources as an infrastructure cloud.
- Enable cloud users to access infrastructure cloud resources more easily.
- Enable scientists and developers to extend and experiment with both sets of capabilities.

The main infrastructure components are the Workspace Service site manager, a web services resource framework (WSRF) based remote protocol implementation, an EC2 based remote protocol implementation of their SOAP and Query APIs, Cumulus, RM API, the cloud client, the reference client, the Workspace Pilot, the workspace-control agent, and the metadata server.

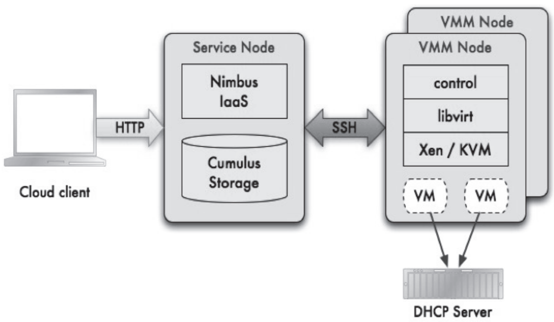

Figure 5: Nimbus Architecture, Source: Nimbus, n.a..

The **Workspace Service** is a site VM manager that different remote protocol frontends can invoke. The **metadata server** responds to HTTP queries from VMs. The **Cloud Client** is the component that enables users to launch instances quickly. The **workspace-control** is a program that can start, stop and pause VMs, implement VM image reconstruction and management, connect the VMs to the network, and deliver contextualization information.

4 COMPARISON OF OPEN SORUCE SOLUTIONS

While the IaaS frameworks discussed in Section III are designed to allow users to manage their own virtual infrastructures, they have different features that should be considered when selecting a framework. This will differ depending on the expected use case and organization. These qualitative features are summarized in Table 1. The set of criteria outlined in Table 1 are derived from a review of the supporting framework documentation and related literature. Based on this review 19 criteria emerged as the main discussed points when comparing open source IaaS frameworks. These criteria are outlined below:

- ***Philosophy:*** Concerns the main use cases for the frameworks deployment (Wen et al., 2012).
- ***Suitability:*** Different frameworks will better service the needs of different users. As a result, large commercial organizations are likely to adopt a different framework than research institutes.
- ***Architecture:*** The architecture of the platform provides details on how the framework was built and how it operates (Mahjoub et al., 2011)
- ***API Support:*** Cloud APIs are application programming interfaces that are used to build applications (Wen et al., 2012; Endo et al., 2010; Semolinski and Thain, 2010).
- ***Amazon Support:*** Given that AWS is the predominant enterprise cloud framework, compatibility with such public clouds is an important factor for many users (Wen et al, 2012).
- ***Cloud Implementation:*** The cloud deployment models (Mell & Grance, 2011) that are suitable for the IaaS framework.
- ***Hypervisor:*** A hypervisor also known as a VM Manager (VMM) is used to describe a technique to allow multiple guests to run concurrently on a

Table 1: Comparison of Open Source IaaS Solutions.

Capability/features	OpenStack	CloudStack	Eucalyptus	OpenNebula	Nimbus
Established	2010	2010	2008	2008	2009
Origin	Rackspace, NASA, Dell, Citrix, Cisco, Canonical etc.	Cloud.com	Santa Barbara university, Eucalyptus System Company	European Union	University of Chicago
Philosophy	Offers Cloud Computing services		Mimic Amazon EC2	Private, highly customizable cloud	Cloud tailored to scientific researchers
Suitability	Enterprises, service providers and researchers	Enterprises, service providers and researchers	Large commercial enterprises, Research institutions	Large commercial companies and public institutions	Research institutions
Architecture	Integration of OpenStack object and OpenStack compute	Hierarchical with four main components: - Management Server, - Availability Zone, - Pod, - Computer Nodes.	Hierarchically grouped from CLC via the CC to the NC ; - Hierarchical - Five components - Minimum two servers	Three modules contain all components; - Centralized - Three components - Minimum two servers	Three modules contain all components; - Centralized - Three components - Minimum two servers
API Support	Native API, Amazon EC2 API, CloudFiles REST API.	Amazon EC2 API, S3	Amazon EC2 API,	Native API in Ruby and JAVA. XML-RPC API for interfaces creation. OGF OCCI & Amazon EC2 APIs.	EC2 APIs, S3 APIs, JAVA client APIs.
Amazon Support	EC2, S3	EC2, S3	EC2, S3, EBS, IAM, AMI	EC2, EBS, AMI	EC2, S3
Cloud Implementation (deployment)	Public Hybrid Private	Public Hybrid Private	Private Hybrid	Private Hybrid	Private Community
Hypervisor	KVM, Xen, VMware ESX, ESXi, Hyper-v, LXC, QEMU, UML, PowerVM, Bare metal	VMware, Oracle VM, KVM, XEN	KVM, Xen, VMware	KVM, Xen, VMware ESX, ESXi	Python, Bash, Ebtables, Libvirt, KVM, Xen
Programming Language	Python	Java	Java, C, Python	Java, Ruby and C++	Java, Python
Community	+++++	++++	+++	+++	++
Release Frequency	<4 months	4 months	>4 months	>6 months	<4 months
Ease of use	+++++	+++++	++	+++	+++
Supported OS	Linux, Windows, Requires x86 Server	Depending on the Hyperviser and hardware - Mac OS X, Asianux, CentOS, Debian, DOS, Fedora, FreeBSD, Novell Netware, Oracle Enterprise Linux, Ubuntu, Red Hat Enterprise Linux, Sun Solaris, SUSE Linex Enterprise, Windows.	Linux (Ubuntu, Fedora, CentOS, OpenSUSE et Debian)	CentOS, Debian, Fedora, RHEL open-SUSE, SLES, and Ubuntu.	Most Linux distributions
Storage	Object and block storage supported. Volumes are persistent (data retained until the volume is deleted, independently from the VM). File storage is supported through Swift (organizing the files in containers).	Supports for iSCSI, NFS, SMB/CIFS; support for OpenStack Swift and Amazon S3	Support for iSCSI, EBS, Amazon S3. Hardware support for industry-standard Storage Hardware.	Hardware support for Fibre Channel, iSCSI, NAS shared storage, SCSI / SAS / SATA. Non-shared and shared file systems (NFS, LVM with CoW, VMFS, etc.).	
Networking	VLAN NO VLAN Public IP's Private IP's SDN IDS Load- balance Firewalls VPN; OpenStack Compute	VLAN, Public IP,	VLAN NO VLAN Public IP's Private IP's; DHCP server on the cluster controller	VLAN NO VLAN Public IP's Private IP's Ebtables OVSwitch; Manual configuration	DHCP server installed on nodes
User Interface	Web interface (i.e. Dashboard) and Command line interface to deploy VMs and a console to manage the VMs.	Web interface and Command Line Interface (CLI)	euca2ools (CLI)	Web interface and Command Line interface (CLI)	Web-Services, specifically: Nimbus Web
Security	API includes protection against DoS attacks or faulty clients. The project concept is introduced by Nova, allowing administrators to manage other user accounts and the project resources. Keystone used for identity management.	CloudStack Secuirty Groups	The Cloud Controller generates a public/private key code pairing for user authentication	Authentication by passwords, secure shell and RSA key code pairings Lightweight Directory Access Protocol ; Authentication framework based on passwords, SSH RSA key-pairs or LDAP. Various administration roles. Multi-tenancy for public clouds.	Public Key Infrastructure
Error Robustness	Replication	Replication	Separate clusters reduce likelihood of correlated errors; Cluster controller's separation	Permanent database to store information about hosts, networks and virtual machines; Database backend (registers virtual machine information)	Regular check and backup of worker nodes; Periodic verification of cloud nodes
Load Balancing	The Cloud Controller	TCP Load Balancer	The Cloud Controller	Nginx	Le Context Broker
Licensing	ApacheLicence Version2	ApacheLicence Version2	BSD-Licence	ApacheLicence Version2	ApacheLicence Version2
Document Support	+++++	+++++	+++	+++	++

host machine (Mahjoub et al., 2011; Wen et al., 2012; Von Laszewski et al., 2010).

- ***Programming Language:*** Concerns the scripting language in which the framework is written.
- ***Community:*** A community may consist of individuals, teams or organizations that are users, developers or service providers. While all may not contribute to the development, the size of the community is a good indication of future developments of the framework (Wen et al., 2012).
- ***Release Frequency:*** The strength of the community will influence the regularity of updates to the open source frameworks. Typically these range from between 4-6 month release cycles.
- ***Ease of Use:*** The complexity of setting up the system and managing VMs. This can be impacted by the supporting documentation available (Wen et al., 2012). Easy of management is crucial for successful implementing Infrastructure as a service (Voras et al., 2013).
- ***Supported OS:*** Concerns the operating systems that are implemented on the machines.
- ***Storage:*** Concerns the support for storage technologies such as network-attached storage, direct-attached storage and backup technologies (Voras et al., 2013). How data is stored on the virtualized pools is critical to achieving flexibility, scaling and ease of use (Wen et al., 2012; Voras et al., 2013).
- ***Network:*** Cloud frameworks must give attention to virtual networks to ensure optimal performance (Voras et al, 2013). Networks can be considered a means of using services deployed on the VMs and managing the cloud environment (Voras et al, 2013; Von Laszewski et al., 2013). This includes support for VLAN, and firewalls.
- ***Interface:*** Refers to how the user will access and manage their resources, typically through a command line interface (CLI) or browser interface (Mahjoub et al., 2011; Wen et al., 2012).
- ***Security:*** Given the perceived security treat of cloud computing, the protection of data is imperative to any IaaS framework Lynn et al., 2014; Mahjoub et al., 2011).
- ***Error Robustness:*** Concerns how the system will tolerate component failure and continue to operate effectively if there is a component failure (Mahjoub et al., 2011).
- ***Load Balancing:*** In computer networking terminology, load balancing denotes how a workload is distributed across multiple machines to optimize resource utilization (Mahjoub et al., 2011).
- ***Licensing:*** Refers to the restrictions of the user on distributing and modifying the software with or without the payment or royalties (Wen et al., 2013). The most common license of open source IaaS frameworks is Apache License Version 2.0 (Openstack, n.a.; Cloudstack, n.a., Von Laszewski et al., 2012).
- ***Documentation Support:*** The level of documentation available to users will impact on the ease of use for setup and management. One of the challenges is keeping documentation updated in line with frequent new releases

5 CONCLUSIONS

In this paper, we present an up to date qualitative review of the main open source infrastructure as a service frameworks. Building on this, we deliver a means of comparison between these frameworks based on a set of criteria developed to evaluate open source frameworks. The paper adds to the literature by providing a comparison of five open source solutions based on the latest releases. We believe that the evaluation will to help potential adopters understand the functions of these frameworks and aid in the early stage adoption decision-making process. This work aims to illustrate the difference offerings for each of the IaaS frameworks and that user requirements may be significantly different, impacting on their framework selection.

In future research, the criteria outlined in this paper will be used to expand this study to compare open and closed IaaS frameworks. We also look to develop performance criteria that can be combined with the existing paper to develop a comprehensive quantitative and qualitative comparison tool.

ACKNOWLEDGEMENTS

The research work described in this paper was supported by the Irish Centre for Cloud Computing and Commerce, an Irish national Technology Centre funded by Enterprise Ireland and the Irish Industrial Development Authority.

REFERENCES

Amazon Web Services Homepage. http://aws.amazon.com/.

Baset, S. (2012). Open Source Cloud Technologies. *Proceedings of the Symposium on Cloud Computing – SoCC'12*, 28-27.

Bist, M., Wariya, M., & Agarwal, A. (2013, February). Comparing delta, open stack and Xen Cloud Platforms: A survey on open source IaaS. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 96-100.

CloudStack Homepage. http://cloudstack.apache.org/.

Cordeiro, T. D., Damalio, D. B., Pereira, N. C. V. N., Endo, P. T., de Almeida Palhares, A. V., Gonçalves, G. E., Mångs, J.-E. (2010). Open Source Cloud Computing Platforms. *2010 Ninth International Conference on Grid and Cloud Computing*, 366–371.

Dukarić, R., & Jurič, M. B. (2013). A Taxonomy and Survey of Infrastructure-as-a-Service Systems. *Lecture Notes on Information Theory*, *1*(1), 29–33.

Endo, P. T., Gonçalves, G. E., Kelner, J., & Sadok, D. (2010, May). A survey on open-source cloud computing solutions. In *Brazilian Symposium on Computer Networks and Distributed Systems*. , 3–16.

Eucalyptus Homepage. https://www.eucalyptus.com/.

Fielder, A., Alleweldt, F., Kara, S., Brown, I., Weber, V., & McSpedden-Brown, N. (2012) Cloud Computing Study. For the European Parliament's Committee on Internal Market and Consumer Protection. IP/A/IMCO/ST/2011-18.

Gartner. (2014). *Magic Quadrant for Cloud Infrastructure as a Service*. Available: http://www.gartner.com/technology/reprints.do?id=1-1UKQQA6&ct=140528&st=sb. Last accessed 30th Sept 2014. Last accessed 30th Sept 2014.

Google Cloud Platform Homepage. https://cloud.google.com/compute/.

IBM IaaS Homepage. http://www.ibm.com/cloud-computing/ie/en/iaas.html.

Keeps, B. (2014). *Cloudstack exces take the money and run or were they pushed.* Available: http://www.forbes.com/sites/benkepes/2014/09/04/on-the-three-year-anniversary-cloudstack-execs-take-the-money-and-run-or-were-they-pushed/. Last accessed 30th Sept 2014.

Kostantos, K., Kapsalis, A., Kyriazis, D., Themistocleous, M., & Rupino de Cunha, P. (2013). Open-Source IaaS Fit For Purpose: A Comparison Between OpenNebula and OpenStack, *International Journal of Electronic Business Management*, 11(3), 191-201.

Kostantos, K., Kapsalis, A., Kyriazis, D., Themistocleous, M., & Rupino, P. (2013). OPEN-source iaas fit for purpose : a comparison between opennebula and openstack, *11*(3), 192–202.

Kumar, R., Gupta, N., Charu, S., Jain, K., & Jangie, S. (2014) Open Source Solution for Cloud Computing Platform Using OpenStack. *International Journal of Computer Science and Mobile Computing,* 3(5), . 89-98.

Lynn, T., O' Carroll, N., Mooney, J., Helfert, M., Corcoran, D., Hunt, G., Van Der Werff, L., & Morrison, J. (2014) Towards a Framework for Defining and Categorising Business Process-As-A-Service (BPaaS), *Proceedings of the 21st International Product Development Management Conference.*

Mahjoub, M., Mdhaffar, A., Halima, R. Ben, & Jmaiel, M. (2011). A Comparative Study of the Current Cloud Computing Technologies and Offers. *2011 First International Symposium on Network Cloud Computing and Applications*, 131–134.

Mell, P., & Grance, T. (2011) The NIST definition of cloud computing. Recommendations of the National Institute of Standards and Technology, Washington, D.C. (NIST Special Publication 800-145).

Microsoft Azure Homepage. https://azure.microsoft.com.

Nimbus Homepage. http://www.nimbusproject.org/.

OpenNebula Homepage. http://www.opennebula.org/.

OpenStack Homepage. http://www.openstack.org/.

Reid, S., Kilster, H., "Sizing the Cloud", Forrester Research Report, 2011.

Ristov, S., Gusev, M., & Donevski, A. (2013). OpenStack cloud security vulnerabilities from inside and outside. In *CLOUD COMPUTING 2013, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*, 101-107.

Salih, N. K., & Zang, T. (2012). Survey and comparison for Open and closed sources in cloud computing. *arXiv preprint arXiv:1207.5480*.

Sefraoui, O., Aissaoui, M., & Eleuldj, M. (2012). Comparison of multiple IaaS Cloud platform solutions. *7th WSEAS International Conference on Computer Engineering and Applications, (Milan-CEA 13). 978-1000.*

Sempolinski, P., & Thain, D. (2010). A Comparison and Critique of Eucalyptus, OpenNebula and Nimbus. *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, 417–426.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009) A break in the clouds: towards a cloud definition, *ACM SIGCOMM Computer Communication Review*, 39(1), 50-55.

Von Laszewski, G., Diaz, J., Wang, F., & Fox, G. (2012, June). Comparison of multiple cloud frameworks. In Cloud Computing *(CLOUD), 2012 IEEE 5th International Conference on,* 734-741.

Voras, I., Orlic, M., & Mihaljevic, B. (2013). The Effects of Maturation of the Cloud Computing Market on Open Source Cloud Computing Infrastructure Projects, *Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces.* 101-106.

Wen, X., Gu, G., Li, Q., Gao, Y., & Zhang, X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, (Fskd), 2457–2461.

Williams, A. (2014). *The Top Open Source Cloud Projects of 2014* Available: http://www.linux.com/news/enterprise/cloud-computing/ 784573 -the-top- open-source-cloud-projects-of-2014. Last accessed 30th Sept 2014.

Yadav, S. (2013). Comparative Study on Open Source Software for Cloud Computing Platform: Eucalyptus, Openstack and Opennebula, *International Journal of Engineering and Science*, 3(10), 51-54.

Zhang, Z., Wu, C., & Cheung, D. W. (2013). A survey on cloud interoperability: taxonomies, standards, and practice. *ACM SIGMETRICS Performance Evaluation Review*, *40*(4), 13-22.