UDS 621.39

A COMPARISON BETWEEN OPEN-SOURCE PLATFORMS FOR CLOUD COMPUTING

Borovskyi B. M.
Institute of Telecommunication Systems,
Igor Sikorsky Kyiv Polytechnic Institute, Ukraine
E-mail: bogdanborovskij@gmail.com

Сравнение открытых программных платформ для облачных вычислений

В этом документе представлен сравнительный анализ платформ с открытым исходным кодом (OpenNebula и OpenStack), которые позволяют предоставлять ресурсы инфраструктуры в качестве службы облачных вычислений. Сравнение проведено на количественной основе - в отношении характеристик и поддерживаемых функций для вышеупомянутых решений с открытым исходным кодом.

Introduction. Cloud computing is rapidly evolving and becoming one of the most challenging paradigms in Information and Communication Technology (ICT) [1]. Many architectural paradigms from distributed computing such as service-oriented infrastructures, Grids and virtualization are incorporated into Clouds. There are three main classes in the cloud services stack:

- Infrastructure as a Service (IaaS);
- Platform as a Service (PaaS);
- Software as a Service (SaaS).

Following the aforementioned main classes in the cloud services stack, in this paper described the IaaS layer by introducing a comparison between two main open-source IaaS solutions. These refer to OpenNebula [2], with more than 30.000 downloads [3] and many companies either contributing (e.g. IBM, Akamai, etc) or using (e.g. Telefonica, Dell, Logica, SARA, etc), and OpenStack [4], with a community of 183 companies (e.g. NASA, Intel, etc). These figures highlight the wide adoption of OpenNebula and OpenStack by not only the research but also the industrial community. A quantitative comparison has been performed in terms of supported functionality and offerings, which are of importance for IaaS solutions. Eucalyptus (yet another open-source solution providing IaaS) as well as proprietary offerings from AWS, Rackspace, GoGrid, Oracle, and other public IaaS providers, are not considered under the scope of this article.

Short overview. OpenNebula is an open-source IaaS solution, initially developed within the framework of an EU-funded project (RESERVOIR). OpenNebula orchestrates storage, network, virtualization, monitoring, and security technologies to enable the dynamic placement of VMs combining both data center resources and remote cloud resources, according to allocation policies.

OpenStack is an open-source IaaS provider project supported by multiple vendors. The project was started by NASA and Rackspace in 2010, while currently more companies have joined (e.g. HP, Dell, Cisco, etc). It is released under the terms of the Apache License, including two sub-projects:

- Nova for computing resources (virtual machines, network, block storage).
- Swift for file-based storage service.

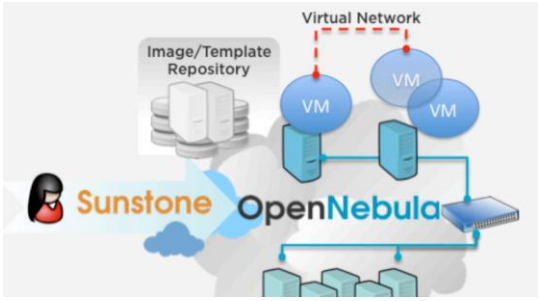

Figure 2: OpenNebula overview

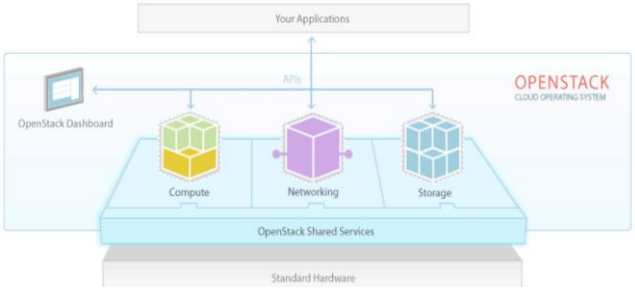

Figure 3: OpenStack overview

The criteria for a qualitative analysis, which has been used to evaluate OpenNebula and OpenStack, has been stated initially. Criteria vary from supported functionality to security. Based on that criteria, the following table (Table 1) summarizes the qualitative comparison analysis between OpenNebula and OpenStack.

Table 1: Qualitative comparison analysis.

Criteria	***OpenNebula***	***OpenStack***
Models supported	Private, public (Amazon EC2 and ElasticHosts), hybrid and federated	Private, public with APIs to Amazon EC2 and S3.
Virtualization supported	Xen, KVM, VMWare ESX, ESXi.	Nova supports: Xen, XCP, QEMU, VMWare ESX, ESXi, KVM, UML.
Access	API, Web interface and Command line interface to deploy VMs and a console to manage the VMs. Web interface also allows image deployment.	API, Web interface (i.e. Dashboard) and Command line interface to deploy VMs and a console to manage the VMs.
Image support	Image repository and creation of image instances (also by saving running VMs) in the catalog. Management of images.	Supported through Glance. Creation is supported through the Amazon EC2 API. Only Linux is supported at the moment.
Resource selection	Enabled through a scheduler that maps requirements to resources.	Not available.
Storage support	Hardware support for iSCSI, FibreChannel, NAS shared storage, SCSI/SAS/SATA. Non-shared and shared file systems (NFS, LVM with CoW, VMFS, etc).	Object and block storage supported. Volumes are persistent (data retained until the volume is deleted, VM independently). File storage is supported through Swift (organizing in containers level).

Criteria	*OpenNebula*	*OpenStack*
Auto-scaling support	Supported – also supports MySQL and SQLite.	Not available.
High-availability support	Persistent storage as a placeholder for the management information. Supports automated testing for scalability, performance and robustness.	Both for core services and some external ones (e.g. MySQL, RabbitMQ).
Monitoring capability	Monitoring of physical and virtual infrastructure.	Infrastructure monitoring available through Swift (reporting to Statsd / Graphite)
API support	Native API in Ruby and JAVA. XML-RPC API for interfaces creation. OGF OCCI & Amazon EC2 APIs.	Native API, Amazon EC2 API, CloudFiles REST API.
Security features	Authentication framework based on passwords, SSH RSA key-pairs or LDAP. Various administration roles. Multi-tenancy for public clouds.	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.
Service Level Agreements (SLA) capability	Enabled through ecosystem tools (such as Claudia and VEP).	Enabled through ecosystem tools (such as Mirantis).

Conclusion. Based on the evaluation of the obtained results from a qualitative perspective, OpenStack is optimum for application requiring rapid resource scaling. Such applications may refer to soft real-time applications (such as Virtual and Augmented Reality, Digital Film Postproduction, etc.) and to applications that may depict spikes in the requested resources - usually the ones that are open to wide user groups (e.g. web applications). On the other hand, OpenNebula would be preferable for applications that pose high elasticity requirements (not supported by OpenStack) such as smart city scenarios (for example enhanced car cruise control by on-demand obtaining and analyzing information from city sensors, other cars, smart phones, etc.) or healthcare scenarios given the OpenNebula security features.

References

1. Buyya, R., Yeo, C. S. and Venugopal, S., 2008, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," 10th IEEE International Conference on High Performance Computing and Communications.
2. OpenNebula, http://www.opennebula.org.
3. Ignacio Llorente, 2012, Cloud Computing Architecture with OpenNebula, NASA Ames.
4. OpenStack, http://www.openstack.org.