

A View of OpenStack: Toward an Open-Source Solution for Cloud

Stanley Lima^{1,2}(✉) and Alvaro Rocha²

¹ Department of Computer Science, Technische Universität Dresden,
01069 Dresden, Germany

`stanley.lima@mailbox.tu-dresden.de`

² Department of Informatics Engineering, University of Coimbra,
3030290 Coimbra, Portugal

`amrocha@uc.pt`, `stanleylima@dei.uc.pt`

Abstract. This paper provides an overview on OpenStack for business leaders and researchers non-experts to identify, distill and easily understand the fundamental aspects of how at present, OpenStack is being developed under the control of the OpenStack Foundation, and how this new platform influences the cloud computing market. Throughout this article we will use a statistical approach to present as OpenStack was built. With this we intend to demonstrate statistically how was the evolution of OpenStack and what are the biggest contributors. For statistical analysis we use the Stackalytics tool that allows tracking of commits and lines of code directly from GitHub, and as a result we achieved a systematic review since the birth of the OpenStack project until the present time. This longitudinal study can be used to examining the variations and changes in the characteristics and the different historical releases of OpenStack project (Kilo, Juno, Icehouse and so on) over the years.

Keywords: OpenStack · OpenStack statistics · Open source software · Cloud computing infrastructures · Scheduling cloud resources

1 Introduction

The National Institute of Standards and Technology (NIST) describes cloud computing, as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

Cloud Computing has changed the IT industry [2], making the software even more attractive through new technologies such as full virtualization or lighter alternatives such as containerization. Nowadays there is a broad discussion on industrial and academic areas about the advantages and disadvantages of various commercial and public cloud technologies such as (IBM, Amazon EC2, Google App Engine, Windows Azure, Hadoop, Eucalyptus, Open Nebula, Nimbus, etc.) [3–5]. However, these commercial clouds usually charge high price

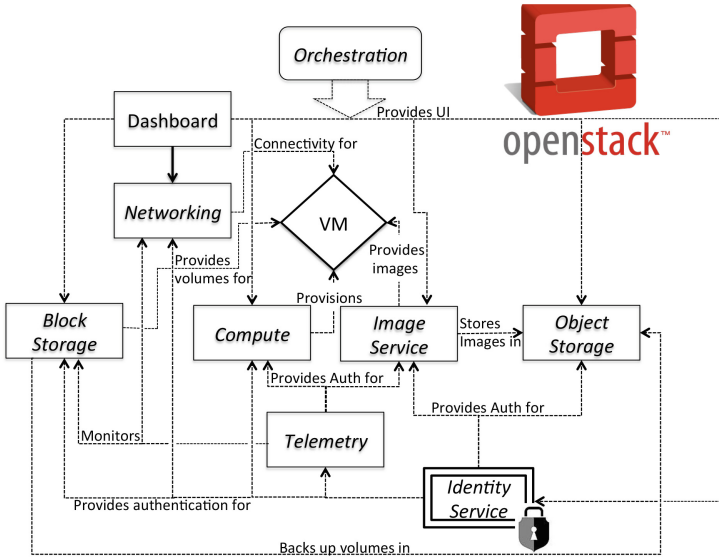


Fig. 1. Conceptual architecture, adapted from Kavanagh [6] 2015.

based on the features that are used. In some specific situations, it is more advantageous for the companies build your own cloud infrastructure. On the other hand OpenStack has emerged as a full-stack of IaaS efficient and through a growing set of open source software platform and under development in addition to being one of the cloud stacks, currently most used. OpenStack is highly configurable to meet different needs. The Fig. 1 shows the relationship between the services of a typical OpenStack environment. We will address each component of this architecture in the next Section.

The remainder of this comprehensive outline is organised as follows: First in Sect. 2, we presented the object of study will be presented in a detailed, updated and statistically descriptive to describe and summarize the open source computing cloud OpenStack project. Then in Sect. 3, community participation in the OpenStack project is presented, which is considered as a major responsible for the growth of this project. Lastly in Sect. 4, the conclusions and recommendations to the issues initially raised by the unfolding of the research will be presented. The objective of this research is not to present open-source cloud management platforms or others infrastructure as a service (IaaS), we will focus only on OpenStack projects.

2 OpenStack

In 2010, Rackspace and NASA jointly launched the first OpenStack distribution titled of Austin [7, 8]. His goal was to create an open-source cloud platform that could provisioning computing services for private and public clouds. The initial

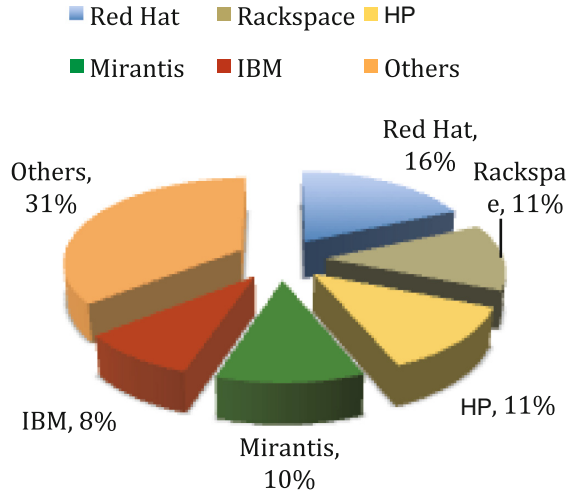


Fig. 2. Contribution by company.

release Rackspace/NASA included only two projects: Swift - for the management object storage system and Nova - for managing compute resource pools.

A way to assess the participation of companies in the OpenStack projects is to look for corporate contributions of all major projects that make up OpenStack. The Fig. 2 shows the share of the companies that most has contributed with commits to OpenStack through the years.

Over all top contributors have been Rackspace, HP, Mirantis, Red Hat and IBM for the past two releases. E.g. the IBM ranked fourth out of 180 companies with more than 4,000 commits to the Newton release. In just one semester, 180 IBM employees contributed with about 14.329 code reviews, implemented 28 blueprints, and fixed 477 bugs for a total of more than 420,000 lines of code.

It is not just about who is the biggest contributor, whether a company wants to invest money, time or human resources for example in an environment or an OpenStack product. Arise relevant questions about potential partners, e.g. In which project the companies are focusing? What is the OpenStack partner really has the expertise to support your production needs? The investment of this partner in OpenStack release is compatible with the products and services that they are selling? This provider could be a long-term strategic partner in your future needs? These are questions that can be answered by analyzing the company’s participation in the OpenStack community.

The OpenStack releases possess different code-names that is proposed and chosen by the technical committee. Each release is comprised of components/services that are interrelated, this modularity enables the deployment of specific services for each case. Each of these services can use each other, it is also possible to modify adapt or customize any service, provided that the basic structure of the APIs is maintained, so that each of the constituent services work

together to provide an integrated solution IaaS. The Fig. 3, shows the relationships between the OpenStack services.

Compute (Nova), is the cloud computing infrastructure manager, all the activities necessary to maintain the life cycle of the instances are controlled by Nova, also is able to manage all resources needed such as cloud scalability, network and so on. The Nova project is a management platform, but does not possess any virtualization capability in itself, it uses the libvirt APIs for supporting a wide variety of virtualization technologies including KVM, Xen, LXC, VMware and more.

Image Service (Glance), provides a image services that lists, records, and retrieves virtual machine images. Glance manage images in a cluster of OpenStack, however is not responsible for the actual storage instead it provides an abstraction for specific storage technologies.

Object Storage (Swift), the OpenStack Object Store project, known as Swift, offers cloud storage software so that applications can store and retrieve lots of data with a simple API. Such objects are stored in different units so Swift ensures that the data is replicated in a clustered server. Initially Swift distributes copies of each object by region (first by region, then by zone, server and across drives).

Dashboard (Horizon), is the implementation of OpenStack Dashboard, which provides a web based user interface to OpenStack services including Glance, Nova, Neutron, Sahara and others. It brings together all OpenStack projects in a single pane based on a Django module called django-openstack.

Identity Service (Keystone), is a OpenStack Identity service that manages user databases and authentication type as (authn) and authorization of high level as (microsoft AuthZ API). It integrates with existing back-end directory services such as LDAP and has recently been re-architected to allow for expansion to support proxy services and external AuthZ/AuthN mechanisms as oAuth, SAML and openID.

Networking (Neutron, formerly Quantum), provides networking as a service to other OpenStack components (such as Nova). It does this by creating and attaching the virtual switch port to the vNIC of the VM. Neutron also includes a set of API interfaces and plug-ins that enable interoperability and network devices orchestration in IaaS environments.

Block Storage (Cinder), implement services and libraries to provide on demand, self-service access to Block Storage resources. Provide Software Defined Block Storage via abstraction and automation on top of various traditional back-end block storage devices, under the most common scenario, the Cinder volumes provide persistent storage to guest virtual machines.

Orchestration (Heat), it is an OpenStack project specifically dedicated to orchestration, making it possible to manage multiple instances in an automatic way. Since the *heat-engine* component provides the main orchestration functionality and supports an orchestration engine to create multiple composite cloud-based model applications in the form of text files that can be treated as

code through a native OpenStack API RESTful as the *heat-api-cfn* component similar to the AWS CloudFormation compliant API Query.

Telemetry (Ceilometer), currently, the telemetry project provides a set of functionality split across multiple projects, each project designed to provide a discrete service in the telemetry space through the following capabilities: Publishes collected data to various targets including data stores and message queues among others.

Database (Trove), is database as-a-service for OpenStack, Trove lays the groundwork for automating the provisioning and management of leading the features of a relational or non-relational database without the burden of handling complex administrative tasks. Initially, the Trove project will focus on resource isolation at high performance while automating complex administrative tasks including patching, backups, restores, configuration and deployment.

Elastic Map Reduce (Sahara), sahara project aims to provide users with a simple means to provision data processing such as Spark or Hadoop on OpenStack. This is accomplished by specifying configuration parameters such as the framework version, cluster topology, node hardware details and more.

Bare Metal Provisioning (Ironic), is the OpenStack service which provides the capability to provision bare metal servers instead of virtual machines, forked from the Nova baremetal driver but Ironic also supports vendor-specific plugins which may implement additional functionality. For a general overview please see: Introduction to Ironic in which its drivers support a variety of hardware vendors such as iBOOT, DRAC, SMTP and others.

Multiple Tenant Cloud Messaging (Zaqar), is the OpenStack Messaging and Notification service. It is similar to Amazon's SQS product with additional semantics to support event transmission. E.g., when installing *python-zaqarclient* the application will have access to the plugin to *python-openstackclient* to create single-row operations. Thus, users can customize Zaqar to achieve a wide range of performance, durability, availability and more efficient targets.

Shared File System Service (Manila), is a community-driven project that presents the management of file shares such as NFS as a core service to OpenStack.

DNSaaS (Designate), provides DNS-as-a-Service for OpenStack such as REST API for domain, record management, integration with Keystone for authentication among others services. It is equivalent to Amazon Route 53. It is authenticated through the Keystone Token, and Designate processes RPC requests through MQ. Uses an API for the DNS Backend, such as MySQLBind and PowerDNS.

Search Service (Searchlight), it aims to improve a search capabilities and performance of various OpenStack cloud services. It accomplishes this by offloading user search queries from existing API servers and indexing their data into Elasticsearch. Elasticsearch is a search server based on Apache Lucene, your goal is to achieve a high performance and flexible query combined with p indexing in real time.

Security API (Barbican), is a REST API OpenStack project designed for the safe storage, provisioning and management of secrets such as passwords, encryption keys and SSL Certificates.

3 Community

In less than five years, OpenStack has become the largest project in open source cloud. Currently has a foundation that gathers some of the biggest IT companies in the world [9] over 180 companies contributing and more than 6,000 individual members [10] in 76 countries. The Fig. 4 shows an analysis of the nine countries among the 170 most active around the world in the OpenStack project.

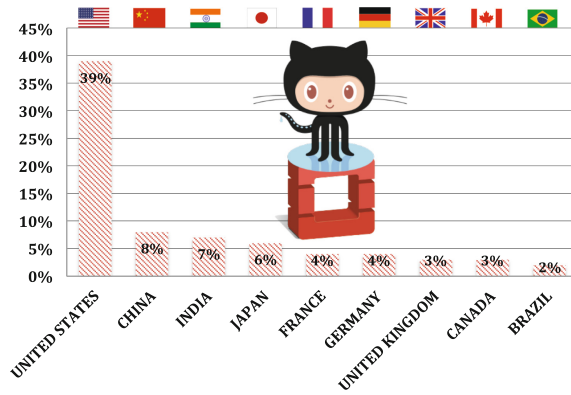


Fig. 4. OpenStack Demographics.

The OpenStack community is extremely active and maintains several options for information about the project, such as:

- OpenStack Maintains a system of meetings between members of the community in IRC Channel, or (Internet Relay Chat) in <https://wiki.openstack.org/wiki/IRC>. It is used as a means of communication in real time between the several OpenStack projects with approximately 82 channels. Some meetings are not held on IRC and are listed here, <https://wiki.openstack.org/wiki/Meetings>.
- OpenStack Docs at <http://docs.openstack.org/host> official documentation for each project and some formations/training program as OpenStack IaaS Exper; AWcloud OpenStack; Firefly’s LaunchPad, Among others.
- Openstack Community at <http://www.openstack.org/community/>, it is a group of people who share a common interest in the OpenStack projects, nowadays has over 30,000 members in over 170 different countries.

- Mailing lists for OpenStack (openstacklists.openstack.org) at https://wiki.openstack.org/wiki/Mailing_Lists detailed lists with different directions and target audience, where each participant may post up to 100 help request messages a day.
- OpenStack blog at <http://planet.openstack.org/> is ever-changing of information discussing since the technical committee updates to events and weekly community newsletters.
- OpenStack-announce at <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-announce> is a list important announcements about OpenStack new releases, stable releases and security advisories.
- Github OpenStack at <https://github.com/openstack/> host all the code developed for OpenStack projects and is available in the public repository from the Rackspace <http://www.rackspace.com/>.
- OpenStack Event Listing at <https://www.openstack.org/community/events/>, displays all past current and the upcoming events on OpenStack.
- OpenStack Ambassadors at <http://superuser.openstack.org/about> is used to share knowledge, experiences and collaborative resolutions of problems between the community members. The topics range from the presentations of case studies to broader issues such as management applications.

The architecture is still modular [11], but as shown in Fig. 5, it has grown at each release by adding new components that provide services in IaaS layer. Nowadays OpenStack is organized in several projects, which have a size and very different levels of activity [12, 13]. Using this thought, we can look for collaborative OpenStack contributions. A basic way of looking at the collaborative contributions is to analyze the contributions aggregated to all Openstack projects.

Statistics are fundamental to transform data into information, this data is supplied from the Stackalytics [14]. Stackalytics process three major metrics for OpenStack contribution: Number of modified files, number of commits and number of modified lines. The related statistics are retrieved from the (`git.openstack`) through the following command:

```
git log --pretty="commit_id: '%H' %ndate:
%at %nauthor: %an %nauthor_email:
%ae %nsubject: %s %nmessage: %b %n'" --shortstat M --no-merges.
```

In result we have the output:

```
commit_id: 123abcxpto date: 11042016 author: Karine kaliane
author_email: karine.kaliane@sample.de subject: add object-based
import in the code message: change class 12hd432 Change-Id:
Idc87sd76367238328apr2016 diff_stat: 7 files changed, 290
insertions (+), 307 deletions (-).
```

As regards the evolution of the OpenStack development community in general, we present in this section statistical graphs that tell a story of growth throughout the release cycle of the OpenStack project.

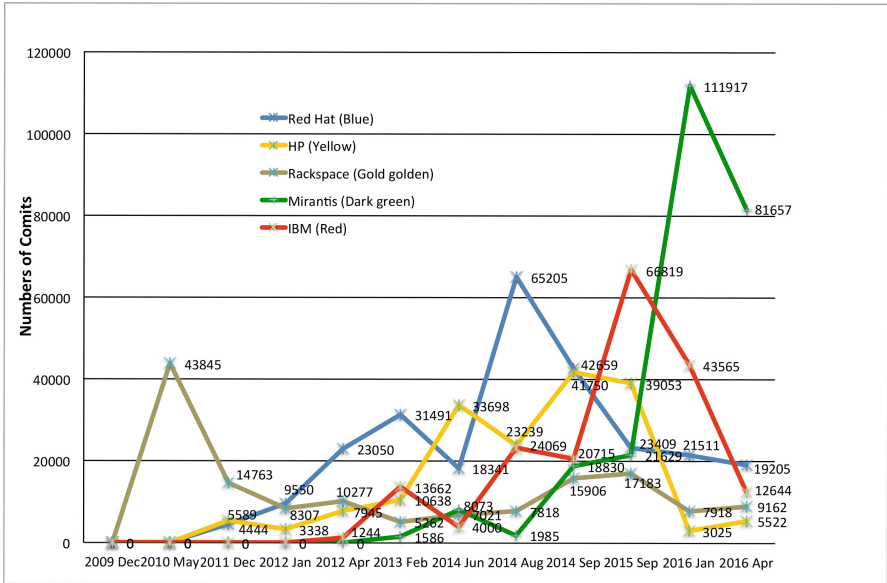


Fig. 5. OpenStack evolution.

In Fig. 5 we can see the commits evolution from 2010 to 2016. It is noticed in the chart that the average is 10000 commits in the years between 2010–2013, increased to 30000 commits in 2015, in 2015 to 2016 grew on average 7.1%, (formula (present) = (passed) * (1 + growth rate)ⁿ. Where n = number of time periods).

This big jump is due to the participation of new companies in the OpenStack project [15]. If we look at the contribution of each company to the latest release of OpenStack, Mitaka, (by number of commits) and express it as a percentage of the total contribution will have the following graph.

Mitaka, the new OpenStack release, shows a continuous increase in activity compared to Liberty. Icehouse to Juno, there was an increase of 7.23 percent of the number of commits. The Juno for Kilo, there is a greater leap in terms of commits (9.23%), but the biggest difference we can see the release version Austin to Diablo and Essex which reaches 32.7% of commits. This progressive increase is directly proportional to the incorporation of new companies to the OpenStack project. Red Hat is the first contributor, with about a quarter of all commits, closely followed by Rackspace, HP and Miratins, and at a certain distance of IBM. Others represents the others totaling more than 550 companies and more than 30,000 members in more than 170 countries collaborating with the OpenStack project [16,17]. Looking at these numbers, it is clear that now the OpenStack project is more leveled, with less dominance by Rackspace, and a clear emerging participation of other companies that appear to be betting big on improving OpenStack codebase.

4 Conclusion

OpenStack is an open platform and scalable and intended for development of public and private clouds with global collaboration among independent developers and specialist companies, such as Huawei and IBM. Although of the newness is warm, the architecture was designed in 2010, in a partnership between Rackspace (American infrastructure provider) and NASA (US space agency), with an initial goal of making the companies consume cloud computing services with elements standardized hardware. In this comprehensive outline, we studied all life cycle of the project, for a better understanding of how OpenStack added value and sold its business model, helping us to see in a simple way the various elements that compose the OpenStack projects.

Acknowledgments. This work was supported by research grants of the programs: Ciências sem Fronteiras (CsF), Agência Espacial Brasileira (AEB), Deutscher Akademischer Austauschdienst (DAAD) and Center for Informatics and Systems of the University of Coimbra (CISUC).

References

1. Mell, P., Grance, T.: The NIST definition of cloud computing (2011)
2. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of (big data) on cloud computing: review and open research issues. *Inf. Syst.* **47**, 98–115 (2015)
3. Wen, X., Gu, G., Li, Q., Gao, Y., Zhang, X.: Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2457–2461. IEEE (2012)
4. Galante, G., De Bona, L.C.E., Mury, A.R., Schulze, B., da Rosa Righi, R.: An analysis of public clouds elasticity in the execution of scientific applications: a survey. *J. Grid Comput.* **14**(2), 193–216 (2016)
5. Eisa, M., Younas, M., Basu, K., Zhu, H.: Trends and directions in cloud service selection. In: 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 423–432. IEEE (2016)
6. Kavanagh, A.: OpenStack as the API framework for NFV: the benefits, and the extension needed. *Ericsson Review* **2**, (2015)
7. Kim, H., Chun, K., Kim, H., Chung, Y.: Utilization of workflow management system for virtual machine instance management on cloud. *Concurrency Comput. Pract. Experience* **27**(17), 5350–5373 (2015)
8. Kumar, R., Parashar, B.B.: Dynamic resource allocation and management using OpenStack. *Nova* **1**, 21 (2010)
9. Qevani, E., Panagopoulou, M., Stampoltas, C., Tsitsipas, A., Kyriazis, D., Themistocleous, M.: What can OpenStack adopt from a ganeti-based open-source IaaS? In: 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), pp. 833–840. IEEE (2014)
10. Campos Plasencia, I., del Castillo, E.F., Heinemeyer, S., Lopez-Garcia, A., Von der Pahlen, F.: Phenomenology tools on cloud infrastructures using OpenStack. *CoRR*, vol. abs/1212.4784 (2012)

11. Van, V.N., Chi, L.M., Long, N.Q., Le, D.-N.: Performance analysis of network virtualization in cloud computing infrastructures on OpenStack. In: Saini, H.S., Sayal, R., Rawat, S.S. (eds.) *Innovations in Computer Science and Engineering*. AISC, vol. 413, pp. 95–103. Springer, Singapore (2016). doi:[10.1007/978-981-10-0419-3_12](https://doi.org/10.1007/978-981-10-0419-3_12)
12. Jones, M.T.: *Cloud computing and storage with OpenStack: discover the benefits of using the open source OpenStack IaaS cloud platform*. Developer Works (2012)
13. Xu, Q., Yuan, J.: A study on service performance evaluation of OpenStack. In: *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, pp. 590–593. IEEE (2014)
14. Stackalytics, April 2016. <http://stackalytics.com>
15. Gonzalez-Barahona, J.M., Izquierdo-Cortazar, D., Maffulli, S., Robles, G.: Understanding how companies interact with free software communities. *IEEE Softw.* **5**, 38–45 (2013)
16. Omar, I.Y., Laborde, R., Wazan, A.S., Barrere, F., Benzekri, A.: G-cloud on OpenStack: addressing access control and regulation requirements. In: *2015 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6. IEEE (2015)
17. Scharf, M., Stein, M., Voith, T., Hilt, V.: Network-aware instance scheduling in OpenStack. In: *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6. IEEE (2015)