

Cloud Computing Frameworks

Quality Model Analysis

Jose Alberto Navas Guerrero (Janague)
Project Evaluation
Máster's on Free Software
Universidad Rey Juan Carlos
Spain
Professor: Daniel Izquierdo Cortázar

January 20, 2014

Abstract

The main purpose of this analysis is to use the Business Readiness Rating (BRR) model to offer a trusted and open framework for Cloud Computing Framework evaluation. In the first phase the students have chosen one of the four free software cloud computing software: OpenNebula, OpenStack, Eucalyptus, and CloudStack. Besides, identify together a list of categories and metrics to be evaluated: Functionality, efficiency, support, documentation, and community. This model aims to accelerate the software evaluation process with a systematic approach, facilitate the exchange of information between students, result in better decisions, and increase confidence in high quality open source software. After the software evaluation for each student, using different role of IT manager have to decide with the best cloud computing framework to company requirements.



Contents

List of Figures

1 Introduction

Nowadays cloud computing technologies are principal to implement services public and private in companies, the major models of cloud computing service are known as Software as a Service, Platform as a Service, and Infrastructure as a Service. These cloud services may be offered in a **Public**, **Private** or **Hybrid** network.

In this report it is defined a quality model based on **Business Readiness Rating for Open Source** (OpenBRR) and it is working in group to define categories and metrics, collecting data, processing data, and analyze results. We suppose we are evaluating software as critical task for our corporate IT like a manager. The four projects are

- Apache CloudStack
- OpenNebula
- OpenStack
- Eucalyptus

1.1 Apache CloudStack

Apache CloudStack is *open source software* designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (*IaaS*) cloud computing platform. CloudStack is used by a number of service providers to offer *public cloud services*, and by many companies to provide an *on-premises (private) cloud* offering, or as part of a *hybrid cloud* solution.

CloudStack is a turnkey solution that includes the entire "stack" of features most organizations want with an IaaS cloud: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting, and a first-class User Interface (UI).

CloudStack currently supports the most popular hypervisors: VMware, KVM, XenServer and Xen Cloud Platform (XCP).

Users can manage their cloud with an easy to use Web interface, command line tools, and/or a full-featured RESTful API. In addition, CloudStack provides an API that's compatible with AWS EC2 and S3 for organizations that wish to deploy hybrid clouds.



Original author(s)	Cloud.com, Citrix
Developer(s)	Apache Software Foundation
Stable release	4.2.0 / October 1, 2013; 3 months ago
Written in	Java (primarily)
Operating system	Cross-platform for management servers, GNU/Linux for hosts, Windows, GNU/Linux, and BSD for guests (depending on hypervisor). ^[1]
Platform	Java
Type	Cloud computing
License	Apache License 2
Website	http://cloudstack.apache.org/ 

Figure 1: CloudStack - Summary

1.2 OpenNebula

Developer(s)	OpenNebula Community
Initial release	March 1, 2008; 5 years ago
Stable release	4.4 🔗 / 3 December 2013; 47 days ago
Preview release	3.7.80 🔗 / 9 October 2012; 15 months ago
Written in	C++, C, Ruby, Java, Shell script, lex, yacc
Operating system	Linux
Platform	Hypervisors (Xen, KVM, VMware)
Available in	English, Russian, Spanish
Type	Cloud computing
License	Apache License version 2
Website	www.opennebula.org 🔗

Figure 2: OpenNebula - Summary

OpenNebula provides the most simple but feature-rich and flexible solution for the comprehensive management of virtualized data centers to enable on-premise IaaS clouds. OpenNebula interoperability makes cloud an evolution by leveraging existing IT assets, protecting your investments, and avoiding vendor lock-in.

OpenNebula can be primarily used as a platform to manage your virtualized infrastructure in the data center or cluster, which is usually referred as *Private Cloud*. OpenNebula supports *Hybrid Cloud* to combine local infrastructure with public cloud-based infrastructure, enabling highly scalable hosting environments. OpenNebula also supports *Public Clouds* by providing Cloud interfaces to ex-

pose its functionality for virtual machine, storage and network management.

1.3 OpenStack

OpenStack is a global collaboration of developers and *cloud computing* technologists producing the ubiquitous open source cloud computing platform for public and private clouds. The project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution.

Founded by Rackspace Hosting and NASA, OpenStack has grown to be a global software community of developers collaborating on a standard and massively scalable open source cloud operating system. Our mission is to enable any organization to create and offer cloud computing services running on standard hardware.

 openstack [™] CLOUD SOFTWARE	
Stable release	Havana (2013.2.1) / 17 December 2013; 32 days ago
Written in	Python
Operating system	Cross-platform
Type	Cloud computing
License	Apache License 2.0
Website	openstack.org 🔗

Figure 3: OpenStack - Summary

Corporations, service providers, VARS, SMBs, researchers, and global data centers looking to deploy large-scale cloud deployments for private or public clouds leveraging the support and resulting technology of a global open source community.

All of the code for OpenStack is freely available under the Apache 2.0 license. Anyone can run it, build on it, or submit changes back to the project. We strongly believe that an open development model is the only way to foster badly-needed cloud standards, remove the fear of proprietary lock-in for cloud customers, and create a large ecosystem that spans cloud providers.

1.4 Eucalyptus

Eucalyptus is a *free and open-source* computer software for building Amazon Web Services (AWS)-compatible *private* and *hybrid* cloud computing environments marketed by the company Eucalyptus Systems. Eucalyptus enables pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change. Eucalyptus Systems announced a formal agreement with AWS in March 2012 to maintain compatibility.

The software development had its roots in the Virtual Grid Application Development Software project, at Rice University and other institutions from 2003 to 2008. Rich Wolski led a group at the University of California, Santa Barbara, and became the chief technical officer at the company headquartered in Goleta, California before returning to teach at UCSB.


EUCALYPTUS	
Developer(s)	Eucalyptus Systems, Inc.
Initial release	1.0 May 29, 2008
Stable release	3.3.0.1 (June 21, 2013; 6 months ago) [±]
Written in	Java, C
Operating system	GNU/Linux, can host Linux and Windows VMs
Platform	Hypervisors (Xen, KVM, VMware)
Type	Private and hybrid cloud computing
License	GPLv3 (only), ^[1] with Proprietary relicensing.
Website	www.eucalyptus.com 

Figure 4: Eucalyptus - Summary

2 Methodology

This Business Readiness Rating model is an evaluation framework for free and open source software (FOSS) and is intended to help IT managers assess which free software would be most suitable for their needs. This is a standardized process, that it usually separates the assessment in four phases.

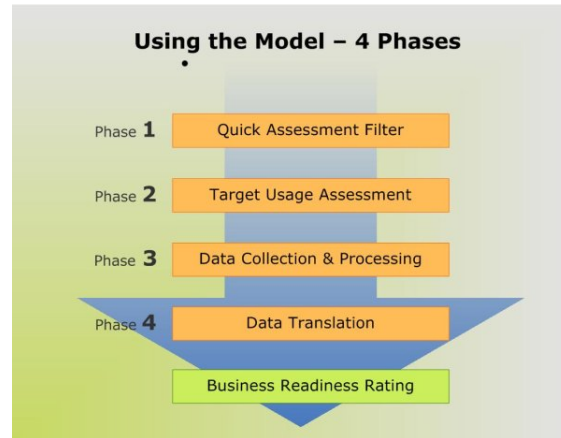


Figure 5: The four phases of software assessment

During the lectures the students groups and professor have chosen the four projects (phase 1), and each student had to choose a particular project, in particular in my case I chose **Apache OpenStack**, and each of us did data collection and processing.

Besides we defined together the metrics of each categories and how to collect and process data. In the OpenBRR methodology

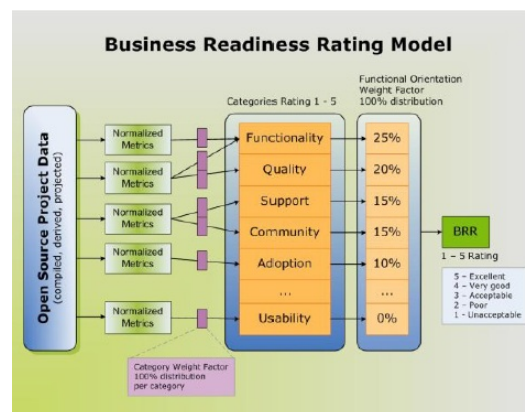


Figure 6: Business Readiness Rating Model

In particular, the role of my imaginary company is a **service provider** with a main objective of implement services in public, hybrid and private clouds. In this context, the most outstanding is to have capacities to serve many functionality, therefore functionality category weight is 30%. On the other hand, our company have a disrupt vision about support, and it would like to have a support given by all members of the community, users, providers, volunteers, and contributor in general. Therefore community categories is as important as functionality, then its weight is 30%.

Evaluated Technology		
Component Name: Apache Cloud Stack		
Component type: Core		
Usage Setting: Cloud Computing Framework		
Rank	Category	Weight
1	Functionality	30.00%
2	Efficiency	20.00%
3	Support	10.00%
4	Documentation	10.00%
5	Community	30.00%
		Total Weight
		100.00%

Figure 7: Quality model: Categories - Weight

2.1 Quality model

In this section we are going to deal with metrics of each category.

2.1.1 Attributes

The lists of attributes and how to calculate were determined in iterative process in class in collaboration with the professor, with objective of being the accurate quality model.

- *Functionality*
 - Billing System
 - Multi Platform Support
 - Administrators Configuration System
 - i18n
 - Quota Facilities
- *Efficiency*

- Performance Testing and Benchmark Reports available
- Performance Tuning & Configuration
- *Support*
 - Company Support
- *Documentation*
 - Documentation Update
 - Number of contributors to documentation
- *Community*
 - Mean commits / developer last month
 - Percent of files touched by only one developer
 - Community growth (commit number variation)

2.1.2 Weight

In the context of service provider the software has at least the following requirements to implement services to its customers: Billing system and quota facilities. Therefore for these metrics the weight is 30%. Another important aspect is administration configuration system in relation with administration and operation of the services, then the weight is 20%.

And the another outstanding aspect is the *community*, we are looking for active, robust and sustainable community; with objective to have the best support. And the context relatively new cloud computing products we are looking for community in growth, therefore its metric weight is 50%.

2.2 Tools

- CVSAnalY

- Retrieves and organizes information from source code management (version control) systems. It currently supports CVS, Subversion and git repositories, with Bazaar and Mercurial in the planning.

- Bicho

- Retrieves and organizes information from issue tracking system. Currently it supports Bugzilla, Jira, and the SourceForge, Allura, GitHub, Google Code, Launchpad, Redmine and GitHub trackers. Trac is in the planning.

- MariaDB

Evaluated Technology		
Component Name:		
Component type: Core		
Usage Setting: Cloud Computing Software		
Rank	Category	Weight
1	Functionality	30.00%
	<i>Billing System</i>	30.00%
	<i>Multi Platform Support</i>	10.00%
	<i>Administrators Configuration System</i>	20.00%
	<i>i18n</i>	10.00%
	<i>Quota Facilities</i>	30.00%
2	Efficiency	20.00%
	<i>Performance Testing and Benchmark Reports available</i>	50.00%
	<i>Performance Tuning & Configuration</i>	50.00%
3	Support	10.00%
	<i>Company Support</i>	100.00%
4	Documentation	10.00%
	<i>Documentation Update</i>	40.00%
	<i>Number of contributors to documentation</i>	60.00%
5	Community	30.00%
	<i>Mean commits / developer last month</i>	25.00%
	<i>Percent of files touched by only one developer</i>	25.00%
	<i>Community growth (commit number variation)</i>	50.00%
		Total Weight
		100.00%

Figure 8: Quality model: Categories - Metrics - Weight

- MariaDB is relational database and a drop-in replacement for MySQL. It strives to be the logical choice for database professionals looking for a robust, scalable, and reliable SQL server. To accomplish this, Monty Program works to hire the best and brightest developers in the industry, work closely and cooperatively with the larger community of users and developers in the true spirit of Free and open source software, and release software in a manner that balances predictability with reliability. To use to analyse the data collected by CVSAnalY

- **OpenBRR template (ods file)**

- **Project Official websites**

- Apache CloudStack - <http://cloudstack.apache.org>
- OpenNebula - <http://opennebula.org/>
- OpenStack - <http://www.openstack.org/>

- Eucalyptus - <https://www.eucalyptus.com/>
- Internet search: Google - DuckDuckGo - wikipedia

2.3 Data sources

The following **source code repositories** have been used

- Eucalyptus: <https://github.com/eucalyptus>
- OpenNebula: <https://github.com/OpenNebula>
- OpenStack: <https://github.com/openstack>
- CloudStack: [git://git.apache.org/cloudstack.git](https://git.apache.org/cloudstack.git)

Project Official websites - Documentation

Internet search: Google - DuckDuckGo - wikipedia

Besides, all dump of databases could be found in <https://github.com/MSWL-PROJ-EV-2013-2014/projev/databases/>

3 Analysis

The Apache CloudStack cloud computing software was analyzed in this section.

Functionality In the web site of Apache CloudStack we can find these features to help to score these metrics.

Metric Name : *Billing System*

Usage Metering Be able to deliver real-time visibility, produce historical usage reports, and create billing invoices by integrating with metering data generated by the CloudStack Usage Engine.

Score: Yes, fully supported

Metric Name: *Multi Platform Support*

Operating system Cross-platform for management servers, GNU/Linux for hosts, Windows, GNU/Linux, and BSD for guests (depending on hypervisor).

Score: No

Metric Name: *Administrators Configuration System*

Domains and Delegated Administration The support for domains allows you to segregate your cloud resources and delegate the management of these domains to other administrators.

Snapshot Management To help mitigate data loss and a way of disaster recovery, users are given the capability of either taking ad hoc snapshots of their disk volumes or setting up a comprehensive schedule for taking periodic snapshots of their data. Users can set limits to the number of snapshots taken to creating new data disk volumes from existing snapshots. Comprehensive Service Management Administrators can define, meter, deploy, and manage multiple service and disk offerings within the cloud. Notification & Capacity Thresholds

Admin can set notification threshold to receive an alert when resource pool crosses the notification threshold. Capacity threshold prevents CloudStack from allocating further resources to that pool.

Score: Yes, a web framework

Metric Name: *i18n*

About documentation, users all complain about documentation, and CloudStack docs is not perfect.

Score: Yes, it is adapted to more than 1 language, but less than 5 languages

Metric Name: *Quota Facilities*

Dynamic Workload Management. Completely automates the distribution of compute, network, and storage resource across the physical infrastructure while adhering to defined policies on load balancing, data security, and compliance. Resource Over Provisioning and Limits. By leveraging existing hypervisor, storage, and network technologies, you can configure the cloud to allow over provisioning of physical resources to optimize the allocation of your virtual machines. CloudStack also provides administrators the capability of limiting virtual resources (such as the number of VMs that an account can create to the number of public IP addresses an account can own).

Score: Yes, for computing, storage and networking

Efficiency .

Metric Name: *Performance Testing and Benchmark Reports available*

There are any performance testing done and benchmarks published, but not many and in general with openStack only.

- Stack wars: OpenStack v. CloudStack v. Eucalyptus - <http://www.networkworld.com/supp/2013/enterprise3/060313-ecs3-open-stack-269899.html>
- OpenStack vs CloudStack: The Latest Score - <http://talkincloud.com/cloud-computing-management/openstack-vs-cloudstack-latest-score>
- OpenStack vs. CloudStack: What's the best approach? - <http://searchnetworking.techtarget.com/tip/OpenStack-vs-CloudStack-Whats-the-best-approach>

Score: Yes

Metric Name: *Performance Tuning & Configuration* There is any documentation and tool to help fine-tune the component for performance, but not many.

- Performance Test Execution for 4.2 - <https://cwiki.apache.org/confluence/display/CLOUDSTACK/Performance+Test+Execution+for+4.2>
- CloudStack Performance Testing by Sowmya Krishnan - <http://open.citrix.com/cloudstack-collab-conference-north-america-2013/video/cloudstack-performance-testing-by-sowmya-krishnan.html>
- CloudStack Performance Testing - <http://www.slideshare.net/buildacloud/cloud-stack-performance-testing>

Score: Yes, Some

Support .

Metric Name: *Company Support* Although there are references to two companies, it seems that only ShapeBlue Inc. gives official support to CloudStack.

- ShapeBlue - <http://shapeblue.com/cloudstack-infrastructure-support/>
- SolidFire - <http://solidfire.com/solutions/cloudstack/>
- Apache CloudStack Matures With First Commercial Support Offering - <http://finance.yahoo.com/news/apache-cloudstack-matures-first-commercial-125600922.html>

Score: Just one company providing support

Documentation .

Metric Name: *Documentation Update* To decide the last time that documentation was updated, we were the update of files with ".txt" extension (it is not really accurate, but it is the first approximation)

The following query was used

```
SELECT
    MAX(s.date)
FROM
    files ff,
    scmlog s,
    file_types f,
    actions a
WHERE
    s.id=a.commit_id and
    f.file_id=a.file_id AND
    f.type='documentation' AND
    ff.file_name LIKE '%.txt'
ORDER BY s.date;
```

The result was on 2014-01-05

```
+-----+
| MAX(s.date) |
+-----+
| 2014-01-03 11:09:24 |
+-----+
1 row in set (5.98 sec)
```

Score: Documentation was updated last week (5)

Metric Name: *Number of contributors to documentation*

Amount of people who contributed to documentation on last year

Query used:

```

SELECT      COUNT(DISTINCT(p.email)) as "Number of Committers"
FROM        scmlog s ,
            people p
WHERE       YEAR(s.date)='2013' AND MONTH(s.date)='11' AND s.committer_id=p.id;

```

Result:

Number of Committers
43

1 row in set (0.04 sec)

Score:10 or more people

Community .

Metric Name: *Mean commits / developer last month*

Query used:

Number of Committers:

```

SELECT      COUNT(DISTINCT(p.email)) as "Number of Committers"
FROM        scmlog s ,
            people p
WHERE       YEAR(s.date)='2013' AND MONTH(s.date)='11' AND s.committer_id=p.id;

```

Result:

Number of Committers
43

1 row in set (0.04 sec)

Number of Commits:

```

SELECT      COUNT(s.id) as "Number of Commits"
FROM        scmlog s ,
            people p
WHERE       YEAR(s.date)='2013' AND MONTH(s.date)='11' AND s.committer_id=p.id;

```

Result:

Number of Commits
650

1 row in set (0.03 sec)

The mean number of commits per developer last month is 15.11

Score: 10 to 20 commits / developer (4)

Metric Name: Percent of files touched by only one developer

Query used:

Number of files:

```
SELECT
    COUNT(DISTINCT(file_path)) AS "File Number"
FROM
    file_links;
```

Result:

File Number
34002

1 row in set (2.72 sec)

Query used:

File touched only by one person:

```
SELECT
    f.file_path ,
    COUNT(p.email) AS "People touching file"
FROM
    scmlog as s ,
    people as p,
    file_links as f
WHERE
    f.commit_id=s.id AND s.committer_id=p.id
GROUP BY f.file_path
HAVING COUNT(p.email)=1;
```

Result:

...

13843 rows in set (0.72 sec)

The percentage of source code files modified only by one developer is 2.46

Score: 5% (5)

Metric Name: *Community growth (commit number variation)*

Query used:

Number of Committers (2012):

```
SELECT
    COUNT(DISTINCT(p.email)) as "Number of Committers"
FROM
    scmlog s ,
    people p
WHERE
    YEAR(s.date)='2012' AND s.committer_id=p.id;
```

Result:

Number of Committers
71

1 row in set (0.03 sec)

Query used:

Number of Committers (2013):

```
select
    COUNT(DISTINCT(p.email)) as "Number of Committers"
from
    scmlog s ,
    people p
WHERE
    YEAR(s.date)='2013' AND s.committer_id=p.id;
```

Result:

Number of Committers
97

1 row in set (0.04 sec)

The percentage of committers number increase/decrease, calculated as percent difference in the amount of people who committed changes on 2012 compared to 2013 is 36.62%

Score: 25 to 50% (4)

Evaluated Technology And Functional Orientation					BRR
Apache CloudStack					3.395
Component type: One					
Usage Setting: Cloud Computing Software					
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
1	Functionality	30.00%	4.4	1.32	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
2	Efficiency	20.00%	3	0.6	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
3	Support	10.00%	3	0.3	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
4	Documentation	10.00%	5	0.5	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
5	Community	30.00%	2.25	0.675	

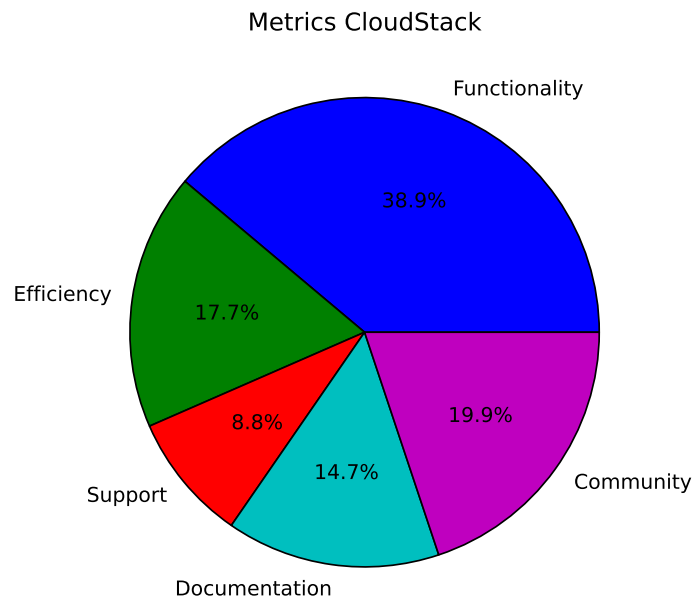


Figure 9: Apache CloudStack Analysis - Score

4 Results

The next phase we were able to use data project from our colleagues with our weights in categories and metrics to do an objective analysis.

Evaluated Technology And Functional Orientation					BRR
OpenStack					3.540
Component type: Nova					
Usage Setting: Cloud Computing Framework					
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
1	Functionality	30.00%	4.6	1.38	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
2	Efficiency	20.00%	3	0.6	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
3	Support	10.00%	5	0.5	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
4	Documentation	10.00%	4.6	0.46	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
5	Community	30.00%	2	0.6	

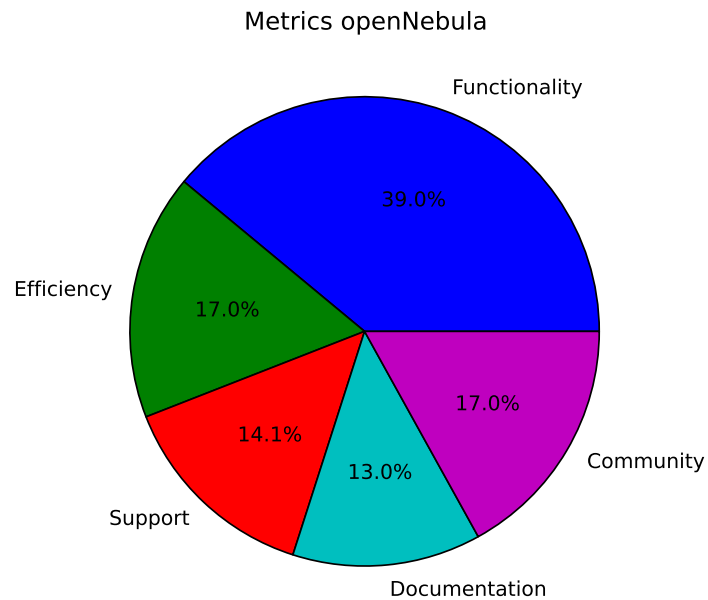


Figure 10: Result - OpenStack - Score

Evaluated Technology And Functional Orientation					BRR	
OpenNebula					3.08	
Component type: One						
Usage Setting: Cloud Computing Framework						
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating		
1	Functionality	30.00%	4.6	1.38		
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating		
2	Efficiency	20.00%	2	0.4		
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating		
3	Support	10.00%	3	0.3		
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating		
4	Documentation	10.00%	4	0.4		
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating		
5	Community	30.00%	2	0.6		

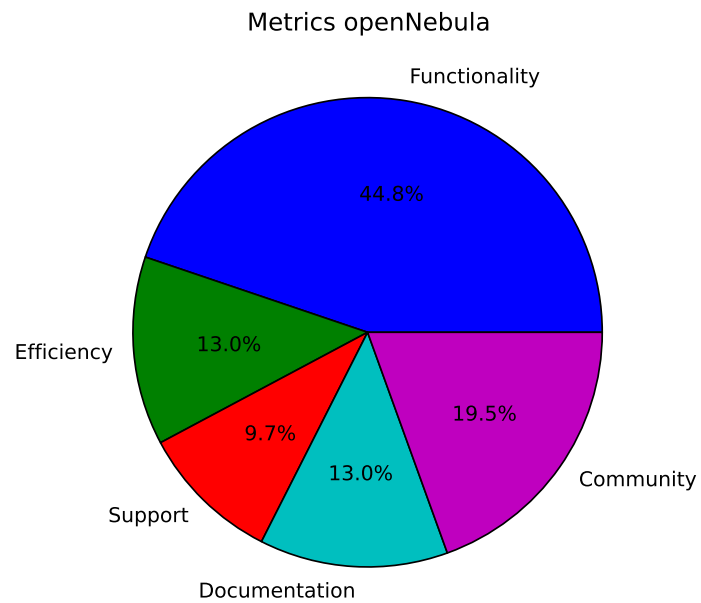


Figure 11: Result - OpenNebula - Score

Evaluated Technology And Functional Orientation					BRR
Component Name: Eucalyptus					3.625
Component type:Eucalyptus					
Usage Setting: Cloud Computing Software					
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
1	Functionality	30.00%	4.8	1.44	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
2	Efficiency	20.00%	3	0.6	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
3	Support	10.00%	3	0.3	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
4	Documentation	10.00%	4.6	0.46	
Rank	Category Title	Weight	Unweighted Rating	Weighted Rating	
5	Community	30.00%	2.75	0.825	

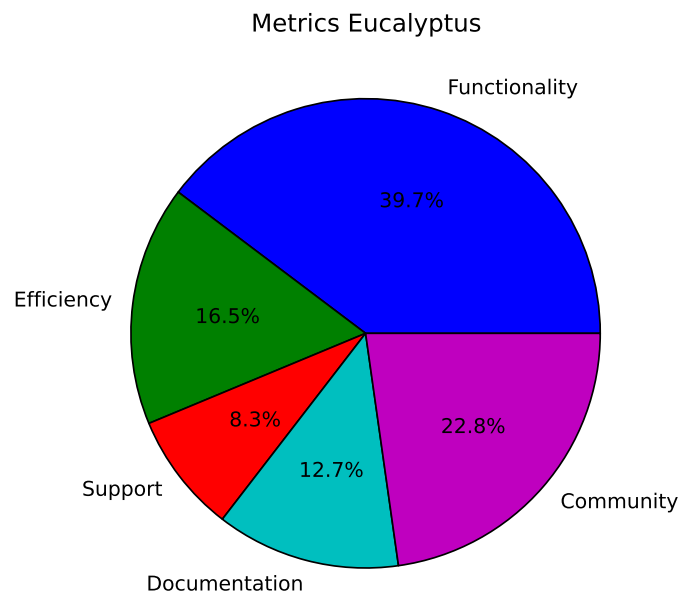


Figure 12: Result - Eucalyptus - Score

As result Eucalyptus is the best election for the role of service provider.

Evaluated Technology And Functional Orientation	BRR
Apache CloudStack	3.395
Eucalyptus	3.625
OpenNebula	3.080
OpenStack	3.540

Figure 13: Result - Final Score

5 Conclusions

After analysis, the Business Readiness Rating (OpenBRR) model offers a trusted, standard and open framework for free software evaluation. In my opinion, it is a fine starting point for a quality model.

First, we defined an equal quality model for every projects, as result Eucalyptus project has the best assessment in my case, with the chosen role and its weights associated with it. This shows the flexibility of the model, with the same rules of analysis it is possible different results.

Before collecting data and processing I had the feeling that OpenStack would be the result, because I know there is outstanding community with a lots contributors.

Maybe our quality model is a good first approximation but it should need iteration to include other metrics and modify some of them. In my opinion, the following list is necessary to thoroughly check.

- With my metrics weight functionality is the key factor, the rest of metrics almost affect to the final score.
- The community metrics do not assess well the size of community, because all the values are relative. Therefore, result are similar in every project, when it is possible to see that OpenStack is the most active community.
- Efficiency metrics are ambiguous.
- We knew that document metrics were not a accurate metric, it should be thoroughly check in a real analysis.
- There would be metrics about licences, because some of the product have private components.
- Include metrics that could be collected automatically would be a improvement.
- Statistics metrics about activity community would be a way to evaluate suitability and robustness of the project.

References

- [1] Apache. "apache cloudstack features". <http://cloudstack.apache.org/software/features.html>, 2013.
- [2] Apache. "apache cloudstack release notes 4.2". http://cloudstack.apache.org/docs/en-US/Apache_CloudStack/4.2.0/html/Release_Notes/index.html, 2013.
- [3] J. M. G. B. J. S. P. G. R. Jordi Mas Hernández (coordinator), David Megías Jiménez (coordinator). *Introduction to Free Software*. UOC, second edition, 2008.
- [4] M. C. Members. "mariadb". <https://mariadb.org/>, 2009.
- [5] openBRR team. "openbrr template". <http://docencia.etsit.urjc.es/moodle/mod/resource/view.php?id=4350>, 2005.
- [6] E. Team. "eucalyptus". <https://www.eucalyptus.com/>, 2013.
- [7] O. Team. "openbrr". <http://www.openbrr.org/>, 2005.
- [8] O. Team. "opennebula". <http://opennebula.org/>, 2013.
- [9] O. team. "opennebula blog". <http://blog.opennebula.org/?p=4372>, 2013.
- [10] O. Team. "openstack". <http://www.openstack.org/>, 2013.