

MSWL Project Evaluation: OpenNebula Project Analysis

Sergio Arroutbi Braojos

OpenNebula
January 8, 2014

(cc) 2014 Sergio Arroutbi Braojos. This work is under a license Creative Commons CC-BY 3.0.

To view a copy of full license, see <http://creativecommons.org/licenses/by/3.0/>

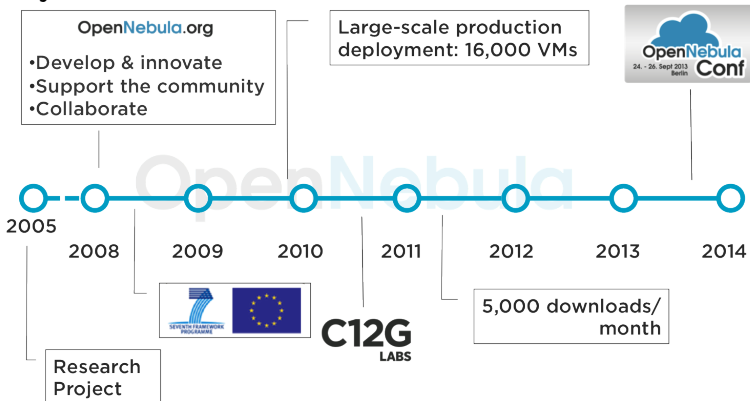


About OpenNebula I

- FLOSS Cloud Computing Framework:
 - Flexible Enterprise Cloud Made Simple
 - Mission: Simple but feature-rich and flexible solution to build and manage enterprise clouds and virtualized data centers
 - Main Objective: Lead innovation in enterprise-class cloud data center management
 - Core Values:
 - 1 Openness
 - 2 Excellence
 - 3 Cooperation
 - 4 Innovation

About OpenNebula II

- Project Start: 2008



Cloud Computing State of the Art I

- OpenNebula
 - CG12 Labs, Focused on Functionality, Weak Community
- CloudStack
 - Open CloudStack, by Apache (Community, Openness)
- Eucalyptus
 - OpenSource AWS Compatible Private Clouds
- OpenStack
 - More than 200 hundred companies involved

Objectives I

"Find which Cloud Computing Open Source Project has more Quality according to an evaluator necessities"

Methodology followed to identify the most appropriate Cloud Computing Open Source Project:

- Candidates Identification (State of Art)
- Define Quality Model (OpenBRR based)
- Invent evaluator role, defining necessities => Weights
- Analyse a project (OpenNebula)
- Compare with competitor projects (OpenStack, Eucalyptus, CloudStack)

Quality Model I

- OpenBRR based. Spreadsheet available.
- Model: 2 Levels. Categories + metrics.
- Categories:
 - 1 Functionality (5 metrics)
 - 2 Efficiency, Benchmarks (2 metrics)
 - 3 Professional Support (1 metric)
 - 4 Documentation (2 metrics)
 - 5 Community (3 metrics)
- Weights: 2 Levels as well:
 - 1 Category weights (Importance of category)
 - 2 Inside category weights (Importance of each metric inside a category)

Role I

- Open Source background
- Community First
- Functionality “comes” with community
- Category weights:
 - 1 Functionality (10%)
 - 2 Efficiency, Benchmarks (15%)
 - 3 Professional Support (20%)
 - 4 Documentation (25%)
 - 5 Community (30%)
- Inside category metric weights:
 - 1 All aspects important inside category
 - 2 No special considerations
 - 3 1 Metric (100%), 2 Metrics (50/50%), 3 Metrics (33/33/33%), 4 Metrics (25/25/25/25%), ...

Analysis I

Tools used:

- OpenNebula Web Portal. Functionality. Professional Support.
- Google. Efficiency.
- cvsanaly2. Documentation and Community.

Data Sources:

- Documentation associated to Key Features on OpenNebula project.
- Documentation associated to Commercial Support.
- OpenStack/CloudStack benchmark comparison.
- OpenNebula “one” Git repository.

Results I

Category	Unweighted Rating	Weight	Weighted Score
Functionality	4.2	10%	0.42
Efficiency	2	15%	0.3
Support	3	20%	0.6
Documentation	4	25%	1
Community	2.33	30%	0.7
Final Score	3.02		

Table: OpenNebula Score

Comparison I

Project	Final Score	Func.	Effic.	Supp.	Doc.	Comm.
CloudStack	3.57	0.38	0.45	0.6	1.25	0.891
Eucalyptus	3.43	0.46	0.45	0.6	1.13	0.79
OpenStack	3.31	0.34	0.45	1	1.13	0.4
OpenNebula	3.02	0.42	0.3	0.6	1	0.7
Winner Project	CloudStack					

Table: Comparison Table

Conclusion I

Is Cloudstack the right solution for this Quality Model?:

- Strengths. Documentation and Community => More weight. Efficiency to a lesser extent.
- Half and Half. Functionality => Less Weight. Worst than Eucalyptus and OpenNebula.
- Weakness. Professional Support => Medium weight. Even here, equals Eucalyptus and OpenNebula.

Summary I

- Identify set of candidates
- Determine a Quality Model
- Set weights (if needed)
- Analyze and classify candidate projects
- Take the final decision

References

- OpenNebula, About OpenNebula Project
<http://opennebula.org/about/project/>
- OpenNebula, OpenNebula Support and Professional Services
<http://opennebula.org/support:support>
- OpenNebula, OpenNebula 4.4 version key features
<http://opennebula.org/about:keyfeatures>
- E. Caron, L. Toch, J. Rouzaud-Cornabas, Comparaison de performance entre OpenStack et OpenNebula et les architectures multi-Cloud:
<http://hal.inria.fr/docs/00/91/69/08/PDF/RR-8421.pdf>
- Metrics-Grimoire, CVSAnalY
<https://github.com/MetricsGrimoire/CVSAnalY>