Project Evaluation

Amal Roumi,

Summary

As a part of my assignment in subject Evaluation of libre software projects, of the Master on libre software of the University Rey Juan Carlos.The main goal of this study is we have to evaluate some of libre software projects, cloud computing system. This report will talk about ..*Eucalyptus* one of the chosen projects.

Contents

List of Figures

List of Tables

1 Introduction

The goals of this research are analyse the different cloud computing in order to know if they match our requirements by provide a comparative report showing metrics and explaining the appropriate degree to these requirements after standardized the process with my classmates. and finally provide references on the data sources, methodology and process followed in this research.

1.1 Cloud Computing

Cloud Computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

Working with a cloud computing has numerous benefits. From a cost perspective, it decreases both the server space and the technical support needed to run and manage complex programs. There are some way need to be considered when selecting one for them and whether there are better alternatives.

Cloud Computing Characteristics

Cloud computing requires when searching for a cloud provider:

- **Elasticity and scalability**
  The cloud is elastic, meaning that resource allocation can get bigger or smaller depending on demand. Scalability also means that an application can scale when adding users and when application requirements change.
- **Self-service provisioning**
  Cloud customers can provision cloud services without going through a lengthy process. You request an amount of computing, storage, software, process, or more from the service provider. After you use these resources, they can be automatically deprovisioned.
- **Standardized interfaces** Cloud services should have standardized APIs, which provide instructions on how two application or data sources can communicate with each other. A standardized interface lets the customer more easily link cloud services together.
- **Billing and service usage metering**
  We can be billed for resources as you use them,this pay-as-you-go model means usage is metered and you pay only for what you consume.

2 Methodology

2.1 Quality model- Open BRR

For analysing the projects we will use the Business Readiness Rating model **OpenBRR**[1] Models. OpenBRR is intended to help IT managers assess which Open Source software would be most suitable for their needs. Open Source users can also share their evaluation ratings with potential adopters, continuing the virtuous cycle and architecture of participation of opensource.[2]
They have defined 12 categories for assessing software

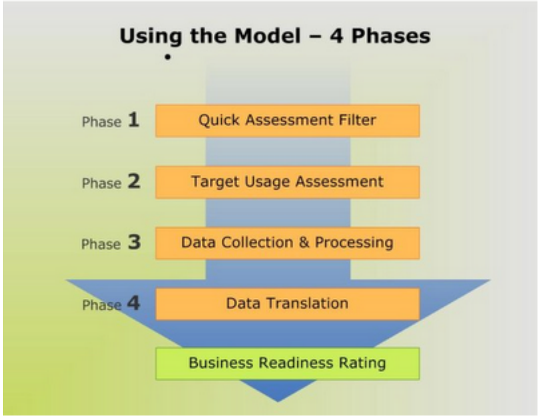
- Functionality
- Usability
- Quality
- Security
- Performance
- Scalability
- Architecture
- Support
- Documentation
- Adoption
- Community
- Professionalism

We are going to follow the the four Phases of **OpenBRR**to apply this model to the evaluation of the cloud computing.

1. Quick Assessment Filter.
   This step starts with a long list of FLOSS projects for quick assessment.
2. Target usage assessment.
   Select the categories that fit our need and rank each metric within a category according to importance to business readiness with total 100% over all the metrics within one category.
3. Data collection and processing.
   Gather data for each metric in each category rating, and calculate the applied weighting for each metric.

[1] http://www.openbrr.org/
[2] http://docencia.etsit.urjc.es/moodle/mod/resource/view.php?id=4343

Figure 1: OpenBRR Four Phases



4. Data translation.
   Use category ratings and the functional orientation weighting factors to calculate the BRR score.

2.2 Tools

Data mining Tools

We used CVSAnalY [3] tool for obtaining related data repositories,developed in python. Its main function is to extracts information out of source code repository logs and stores it into a database with all the information regarding the repository, users, commits, and files. Which can be easily to obtain results.

2.3 Data sources

We will use data from different sources for analysing :

- **Official websites** The official websites of the Euclyptus provide reliable information about it's functionality, documentation and support.
  Eucalyptus: `https://www.eucalyptus.com/`
- **Source code** :`https://github.com/eucalyptus/eucalyptus`
- **Wikis** `https://github.com/eucalyptus/eucalyptus/wiki`
- **Documentation** `https://www.eucalyptus.com/docs`
- **Benchmarking** Some PDF studies listed as references.
- **Repository of Repositories: RoR** are analysis repository tools.Display information about project in an easily way. **Ohloh** is a free public ROR directory of open source software and people.Some metrics present in Ohloh site are provided by Ohloh specific tools and others are provided by Ohloh users. Ohloh owned and

[3] //github.com/MetricsGrimoire/CVSAnalY

operated by BlackDuck Software a consulting company specialized in gathering and providing information about open source software projects.

- **OpenBRR sheets** We will use OpenBRR sheets made by my classmates in Libre Software master in 2013/2014 (Open stack,Cloud stack and Open nebula). May be downloaded from:
  - *Open stack* `https://github.com/MSWL-PROJEV-2013-2014/projev/blob/master/spreadsheet/OpenStack/OpenStack_BRR_ProjEval_2013.ods`
  - *Cloud stack* `https://github.com/MSWL-PROJEV-2013-2014/projev/tree/master/spreadsheet/CloudStack`
  - *Open nebula* `https://github.com/MSWL-PROJEV-2013-2014/projev/blob/master/spreadsheet/OpenNebula/BRR_ProjEval_eucalyptus.ods`

3 Analysis and Results

Applying the **Open BRR** four phases of the projects.

3.1 Phase 1: Quick Assessment Filter

In the Quick Assessment phase, the idea is to quickly eliminate products that are clearly unsuitable and to decide the target usage of the applicationbut in our case the first phase has been applied for the four cloud computing system(Eucalyptus OpenStack,OpenNebula,CloudStack). Before start with evaluation let's take a look at the cloud computing that will be evaluated.

3.1.1 Eucalyptus

Eucalyptus is open source private cloud software for building private and hybrid clouds that are compatible with AWS APIs. With AWS-compatibility, the open source software pools together existing virtualized infrastructure to create private or hybrid cloud resources for compute, network,and storage.
FEATURES[4]

- AWS API Compatibility with EC2, EBS, S3, IAM, Auto Scaling, Elastic Load Balancing, and CloudWatch Management
- Supports AWS Ecosystem
- User Console with Hybrid Cloud
- Role-Based Access Management

[4] https://www.eucalyptus.com/sites/all/files/ds-eucalyptus-iaas.en.pdf

- High Availability
- Flexible Clustering
- Quota Management and Accounting
- Resource Tagging
- Customizable Instance Types
- Maintenance Mode
- Network Management, Security Groups,Traffic Isolation
- KVM Hypervisor Support
- Microsoft Windows and Linux Guest OS

3.1.2 OpenStack

In July 2010 Rackspace Hosting and NASA jointly launched an open-source cloud-software released under the terms of the Apache license initiative known as OpenStack.A cloud-computing project, aims to provide the ubiquitous open source cloud computing platform for public and private clouds, The project is managed by the OpenStack Foundation[5].

3.1.3 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 ,under the terms of the Apache license [6].

3.1.4 Open Nuebla

OpenNebula is free and open-source softwarea cloud computing toolkit for managing heterogeneous distributed data center infrastructures. The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid implementations of (IaaS) under the terms of the Apache license [7]

3.2 Phase 2: Target Usage Assessment

The second phase consists in setting the category and metric weights according to the company's requirements.While the model provides twelve assessment categories the model recommends using no more than seven of these.Once the categories have been selected, an average weighting factor must be set to know how much each category should contribute to the final result, using scoring one to five for each defined metric.

[5] http://www.openstack.org/

[6] http://cloudstack.apache.org/

[7] http://opennebula.org/about/project/

We have chosen a set of metrics to represent the most important values related to cloud computing evaluation
Table and chart below shows the weight that given to each category for our evaluation

Figure 2: Matrics chosen in chart

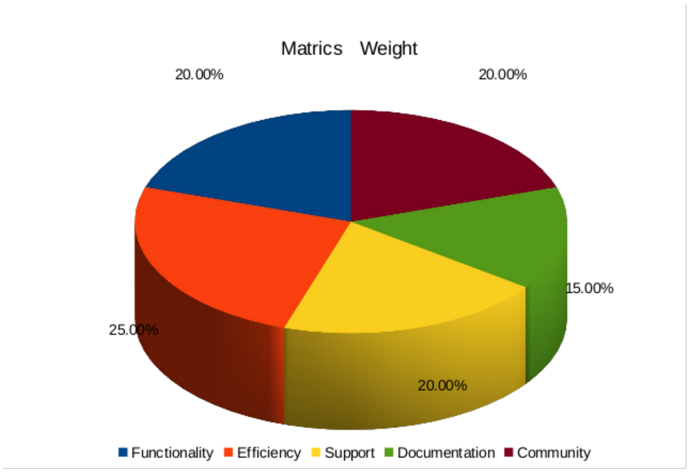


Table 1: OpenBRR matrices chosen

Category	Weight
Functionality	20.00%
Efficiency	25.00%
Support	20.00%
Documentation	15.00%
Community	20.00%
TOTAL	**100%**

We can see that the value that is given more weight is Efficiency with 25 %,then the Functionality, Support and Community 20% and last support with 15%

3.3 Phase 3: Data collection and processing

The third phase is by far the most time-consuming, because all measurement data need to be compared with a normalized scale in order to convert the measurement in something meaningful. The tables below shows metric used in all the categories ,we will refer to Unweighted score as Unweighted and Weighted Score as Weighted.

I added new field to BRR spreadsheet for the reference from were we obtained the metric data and in some cases we will show queries on how we obtained this data by Using the database dump from FLOSSmetrics for Eucalyptus project, we can find the database in this link `https://github.com/MSWL-PROJ-EV-2013-2014/projev/tree/master/databases/eucalyptus`

Table 2: Efficiency

Metric Name	Weights	Unweighted	Weighted
Performance Testing and Benchmark Reports available	50.00 %	3	1.5 %
Performance Tuning & Configuration	50.00 %	3	1.5 %

Table 3: Functionality

Metric Name	Weights	Unweighted	Weighted
Billing System	30.00 %	5	1.5
Multi Platform Support	20.00 %	5	1
Administrators Configuration System	20.00%	5	1
i18n	10.00%	3	0.3
Quota Facilities	20.00 %	5	1

Table 4: Support

Metric Name	Weights	Unweighted	Weighted
Company Support	100%	3	3

Table 5: Community

Metric Name	Weights	Unweighted	Weighted
Mean commits / developer last month	40.00 %	4	1.6
Percent of files touched by only one developer	20.00 %	1	0.2
Community growth	40.00 %	3	1.2

- To calculate the number of commits per developer last month.
  We use this queries :
  Number of commits

```
 mysql> select Count(scmlog.id) from scmlog , people
 where scmlog.committer_id=people.id and
 year(scmlog.date)='2013'
 and month(scmlog.date)='11';
+------------------+
| Count(scmlog.id) |
+------------------+
|              234 |
+------------------+
```

Number of Developers

```
mysql>   select Count(DISTINCT(people.id)) from scmlog , people
 where  scmlog.committer_id=people.id
  and year(scmlog.date)=’2013’ and month(scmlog.date)=’11’;
+----------------------------+
| Count(DISTINCT(people.id)) |
+----------------------------+
|                         15 |
+----------------------------+
```

The result for this is around 16

- Percent of source code files modified only by one developer.

```
mysql> select Count(DISTINCT(file_path)) from file_links;
+----------------------------+
| Count(DISTINCT(file_path)) |
+----------------------------+
|                       8586 |
+----------------------------+
```

```
mysql> select Count(people.email) from scmlog  ,people,  file_links
 where file_links.commit_id=scmlog .id and scmlog.committer_id=people.id
  GROUP BY file_links.file_path HAVING COUNT(people .email)=1;
5239 rows in set (0.34 sec)
```

Files=8586
Developer=5239
= 61.02% of source code files modified only by one developer

- Community growth Percent of commiters number increase/decrease, calculated as percent difference in the amount of people who committed changes on 2012 compared to 2013

```
mysql> select Count(people.id) from scmlog , people
 where year(scmlog.date)=’2012’ and scmlog.committer_id=people.id;
 +-----------------+
| Count(people.id) |
+-----------------+
|             3980 |
+-----------------+
```

```
mysql> select Count(people.id) from scmlog , people
  where year(scmlog.date)=’2013’ and scmlog.committer_id=people.id;
```

```
+-------------------+
| Count(people.id) |
+-------------------+
|             4155 |
+-------------------+
```

To calculate the growth rate = (v2013-v2012)/v2012*100 = 4.4

Table 6: Documentation

Metric Name	Weights	Unweighted	Weighted
Documentation Update	50.00%	4	2
Number of contributors to documentation	50.00%	5	2.5

- Documentation Update When was the last time that documentation was updated

```
mysql>  select max(scmlog.date) from  scmlog , file_types , files ,
actions  where scmlog.id=actions.commit_id and file_types.file_id=actions
 and file_types.type='documentation' and files.file_name like '%.txt'
 order by scmlog.date;
+---------------------+
| max(scmlog.date)    |
+---------------------+
| 2013-12-08 21:05:06 |
+---------------------+
```

- Number of contributors to documentation Amount of people who contributed to documentation on last year

```
mysql> select Count(DISTINCT(people.id)) from files,scmlog ,file_types,
people , actions  where scmlog.committer_id=people .id
and scmlog.id=actions.commit_id and file_types.file_id=actions.file_id
and file_types .type='documentation'
  and files.file_name LIKE '%.txt' and year(scmlog.date)='2013';
+----------------------------+
| Count(DISTINCT(people.id)) |
+----------------------------+
|                         34 |
+----------------------------+
```

3.4 Phase 4: Data translation

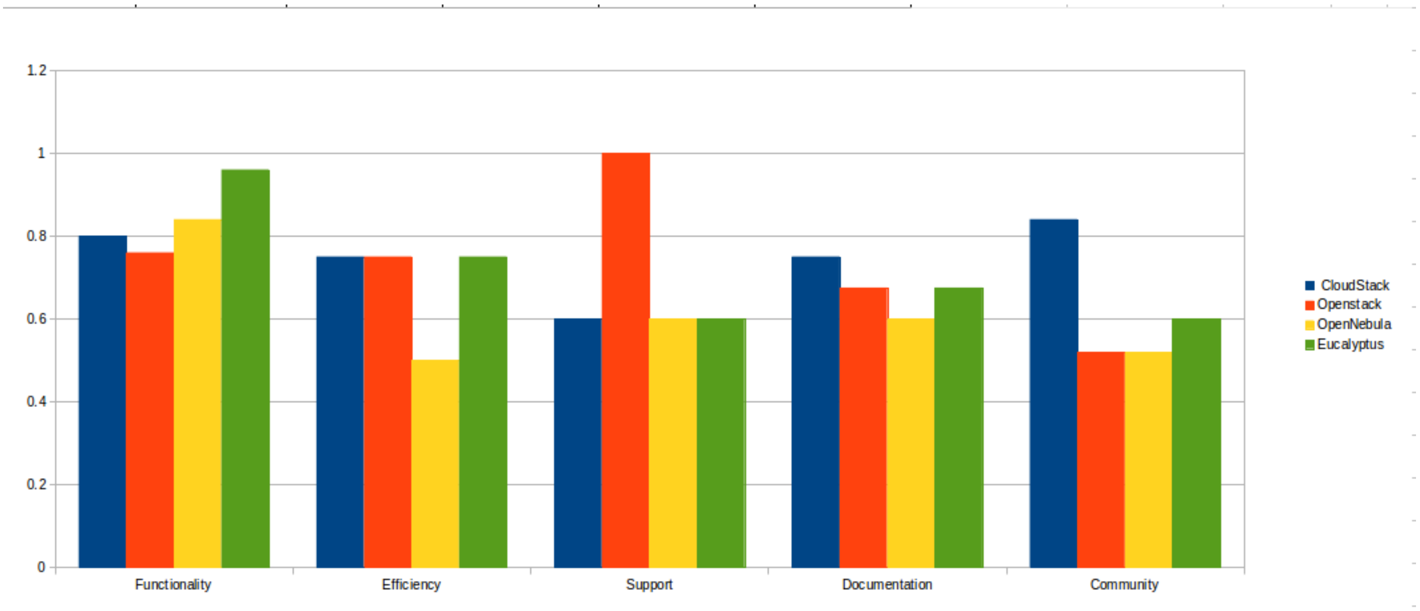
Taking into account the previous normalized data, the data translation phase calculates the final score by computing the ratings and weighted factors using the OpenBRR spreadsheet where the scores for each category is automatically calculated. The resulting work can be downloaded from this URL:
OpenBRR spreadsheet in`https://github.com/MSWL-PROJ-EV-2013-2014/projev/blob/master/spreadsheet/eucalyptus/BRR_ProjEval_eucalyptus.ods`

In the table below we can see a comparison of the results obtained from the analysis of the four cloud computing projects:

Table 7: OpenBRR Comparison of the results

Cloud Computing					
Category	**Weight**	**Eucalyptus**	**Cloudstack**	**OpenStack**	**OpenNebula**
Functionality	20,00%	0.96	0.8	0.76	0.84
Efficiency	25,00%	0.75	0.75	0.75	0.5
Support	20,00%	0.6	0.6	1	0.6
Documentation	15,00%	0.68	0.75	0.68	0.6
Community	20,00%	0.6	0.84	0.52	0.52
TOTAL	**100%**	**3.59**	**3.74**	**3.71**	**3.06**

Figure 3: Cloud Computing comparison



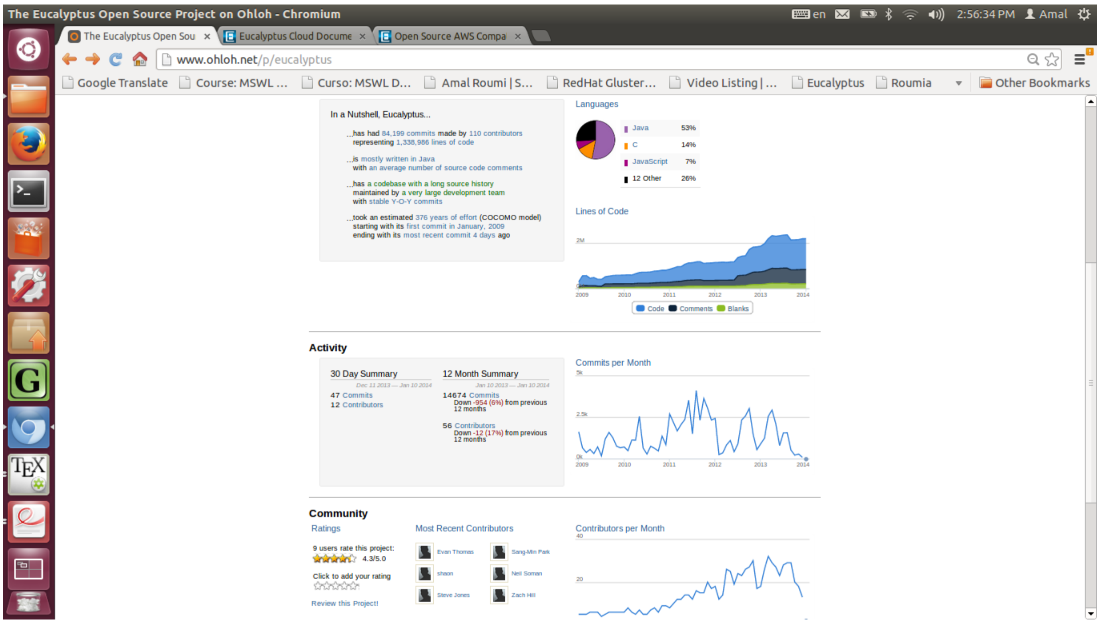
Form the above we can derive the following conclusions:

1. Functionality Eucalyptus offers a good rate that covers most of the functionalities then comes Open Nebula and finally Open Stack
2. Efficiency : In terms of Efficiency and Performances we can see that Open Nebula is not so fast as other Eucalyptus ,Cloudstack and OpenStack which they have the same rate in the Efficiency
3. Support: Support is an important factor, we can see that Open Stack has the highest rate, in our case in the support category more than one company providing professional support.
4. Documentation: We found that the Cloud Stack maintainer and developers care about documentation, not only the source code.
5. Community: Cloud stack has most rating in community ,followed by Eucalyptus and finally Open Stack and Open Nebula.

3.5 Other matrics

Apart from the metrics provided by OpenBRR we have also considered the Ohloh:Figure below shows an example of the evolution of lines of code from Eucalyptus Ohloh statistics:

Figure 4: Line of code in eucalyptus



In a Nutshell, Eucalyptus...

...has had 84,199 commits made by 110 contributors representing 1,338,986 lines of code

...is mostly written in Java with an average number of source code comments

...has a codebase with a long source history maintained by a very large development team with stable Y-O-Y commits

...took an estimated 376 years of effort (COCOMO model) starting with its first commit in January, 2009 ending with its most recent commit 4 days ago

Languages

Java	53%
C	14%
JavaScript	7%
12 Other	26%

Activity

30 Day Summary
Dec 11 2013 — Jan 10 2014
47 Commits
12 Contributors

12 Month Summary
Jan 10 2013 — Jan 10 2014
14674 Commits
Down -954 (6%) from previous 12 months
56 Contributors
Down -12 (17%) from previous 12 months

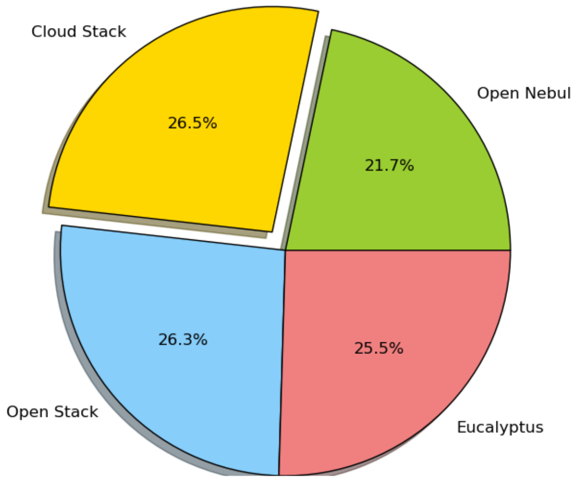
Community

Ratings
9 users rate this project: 4.3/5.0
Click to add your rating
Review this Project!

Most Recent Contributors
Evan Thomas
Sang-Min Park
shaon
Neil Soman
Steve Jones
Zach Hill

4 Conclusion

Figure 5: Cloud Computing result



After analysing the results of the metrics applied to Cloud computing Systems considering the weight we have given to each of the metrics,the total results for each cloud computing are Open Nebula with (3.06),,Eucalyptus with (3.59), Open Stack(3.71) and Cloud Stack with highest rate with (3.74). we can say that: **Cloud Stack** is the winner with 26.5 % .

References

[1] Eucalyptus web-page,
https://www.eucalyptus.com/

[2] ISO/IEC Standard Software engineering,
http://en.wikipedia.org/wiki/ISO/IEC_9126

[3] Measuring the Performance of Open Source Development Communities: The QualOSS Approach,
Daniel Izquierdo-Cortazar, Martn Soto,and Marcus Ciolkowski
http://www.libresoft.es/publications/2009-qualoss-approach-metrikon

[4] OpenBRR,
http://www.openbrr.org/

[5] FLOSS Communities: Analyzing Evolvability and Robustness from an Industrial Perspective
Daniel Izquierdo-Cortazar, Jesus M. Gonzalez-Barahona, Gregorio Robles, Jean-Christophe Deprez, and Vincent Auvray
http://libresoft.es/publications/2010-floss-communities-oss

[6] Towards Automated Quality Models for Software Development Communities: the QualOSS and FLOSSMetrics case
Daniel Izquierdo-Cortazar, Jesus M. Gonzalez-Barahona, Santiago Dueñas, Gregorio Robles
http://libresoft.es/publications/2010-quality-models-quatic

[7] A Performance Comparison of Clouds Amazon EC2 and Ubuntu Enterprise Cloud
http://www.cs.st-andrews.ac.uk/files/PerformanceComparison.pdf

[8] Cloud Computing Performance Benchmarking and Virtual Machine Launch Time.
http://sigite2012.sigite.org/wp-content/uploads/2012/09/p89.pdf

[9] Performance of Eucalyptus and OpenStack Clouds on Future Grid
http://research.ijcaonline.org/volume80/number13/pxc3891831.pdf

[10] PRweb
http://www.prweb.com/releases/2010/08/prweb4423554.html

[11] Matplotlib
http://matplotlib.org/index.html