Approach proposal for competitive analysis

Gathered by Patrick Moreau
INRIA
Software Assets Manager

patrick.moreau@inria.fr

OSCI workshop, Paris, 23/11/2010

Introduction

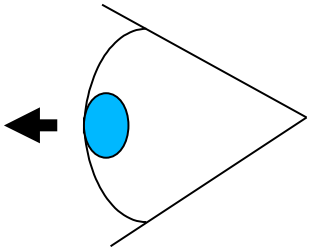
I am up to now alone on this part.

Collaborative work would be preferable for this important (essential?) task of the initiative

A complex analysis has to be done:

- deliverables: advice, recommendations, and finally a roadmap

3 proposals



To establish the motivation to adopt Open source solutions for Cloud

To perform a macroscopic comparison of OSCi stacks with cloud stacks

- open source
- non open source

To perform a brick by brick comparison of OSCi stacks with open source cloud stacks

- "If we don't want that OSCi results look like Frankenstein …"

Motivation for an OS solution

To be completed (sub-task1)

From an user point of view

- data and Application Interoperability (freedom)
- data and Application Portability
- transparency / Privacy / Security

From a technology provider / integrator point of view

- confidentiality / security
- general consensus on reduction of costs and barriers to adoption
- open APIs for pragmatic reasons
- OSS and open standards

Proposal: to write a white paper

macroscopic comparison of OSCi stacks

Open source

- OpenNebula (to put in the radar)
- OpenStack (seems complementary)
- Eucalyptus: no more open source (open core)?
- Nimbus (?)
- Apache (RedHat)
- Open cloud.org (IBM being the leader)
- Rackspace
- Other of interest?

Non open source

- Amazon
- Google
- Amazon EC2,
- GoGrid
- Amazon S3
- Nirvanix
- Linode
- other of interest?

Criterias To be completed (sub-task2)

Confidentiality . Security

- cryptography, traceability
- end to end security

SLA commitments

- quality, continuity of services, SLA management

Access right management

Scalable data base

Compatibility with standards

- OCCI (open cloud computing interface), OVF (Open virtualization format), Nessi

Deployment

- service deployment

Management and orchestration

Service billing

Energy savings

Brick by brick comparison

This comparison has to be performed for each domain:

- Self-sizing & Green PaaS
- Interoperable & Distributed Services
- BI 4 Cloud
- Massively Distributed Clouds (less relevant)

For all of us, who provides bricks:

- How are we sure that what we provide is better than existing components?
- What are the criterias of analysis?
- what are the competitive features?

To be completed (sub-task 3)

Brick by brick comparison

Libraries / APIs: Deltacloud, jCloud, libvirt, libCloud, ...
VM: Abiquo, Convirt, KVM, OpenVZ, Qemu, VirtualBox, Xen, …
Development: JEE, Eclipse, POJO, Spring, Seam, Struts, GWT, Groovy, JRuby, V8, ...
IaaS: ControlTier, Enomaly, Eucalyptus, OpenNebula, NiftyName, Nimbus, OpenStack, OpenQRM, Puppet, RabbitMQ, Reservoir,Traffic Server, Ubuntu, ...
PaaS: Appscale, Gearman, Heroku, Joyent, WaveMaker, …
SaaS: Coadunation, Cornelios, eyeOS, Guacamol, TioLive, ...
Deployment / Admin / Monitoring / Test: Bitnami, Capistrano, CDT, Cfengine,Chef, collectd, Bcfg2, Etics 2, Fabric, ganeti, Maven, Puppet, Zenoss, ...
Storage and NO/SQL: Cassandra, CouchDB, DRDB, Drizzle, Flare, Memcached, MongoDB, Neopod, XtreemFS, ...
File Systems: CloudStore, GlusterFS, Gpfs, Hdfs, Pohmelfs, ...
Auto scalability: Scalr, ...
Data processing: Hadoop, MapReduce, Pig Zookeeper, ...
Green IT / Smart Grid: Nedo, …
Billing: Jbilling, ...