

Cloudware support for Scientific Applications

Dana Petcu

West University of Timisoara
& Institute e-Austria Timisoara
Romania

web.info.uvt.ro/~petcu



Sequoia/DOE



Google Compute Engine: For \$2 million/day, your company can run the third fastest supercomputer in the world

By Sebastian Anthony on June 28, 2012 at 3:18 pm | [3 Comments](#)



Share This Article

 128	 53	 20	
 Like	 Tweet	 2,613	 reddit

At the Google I/O conference in San Francisco, Google has announced the immediate availability of Compute Engine, an infrastructure-as-a-service (IAAS) product that directly competes with Amazon EC2 and Microsoft Azure. Citing more than a decade of

running and optimizing its own data centers and network infrastructure, Google is claiming that the Compute Engine is more scalable, more stable, and cheaper than the competition.

For this story, we'll focus on scalability and cost (I'm sure that Compute Engine is stable, but Google just hasn't given us any figures to work with). Google says that Compute Engine has access to 770,000 cores — a figure that will surely grow over time. In one demo at Google I/O, a genomics app (it analyzed the human genome) was shown to use 600,000 cores. These cores are made available as Linux virtual machines (VMs), with 1, 2, 4, or 8 cores each. Each core apparently has access to 3.75GB of RAM each — and, of course, each VM is connected together using Google's advanced networking technologies and topologies.

777,000 cores, assuming the entire Compute Engine cluster consists of 8-core CPUs, equates to 96,250 computers. This is a huge number — probably equal to the total number of servers operated by Intel, or data centers such as The Planet or Rackspace, but

Content

- Advantages & disadvantages of Cloud for science
- One support tools



Cloud computing for scientists?

- **Facts about Cloud computing:**

1. Emerged from business reasons
2. Serves mainly web applications
3. Technologies are still evolving
4. Still 'trendy'

- **Delivery models**

- Infrastructure (IaaS), Platform (PaaS), Software (SaaS)
- other XaaS

- **Questions:**

1. Which are the expectations scientific applications from CC?
2. Which are the proof-the-concept applications?
3. Status of the current support for sci applications from PaaS?



Expectations of the scientist from an execution environment

1. availability of scientific libraries and domain specific tools;
2. usage of common programming languages;
3. storage capacity for large data;
4. sharing data and codes with the fellow researchers;
5. security against the public eyes;
6. easy deployment of new codes;
7. high performance hardware at no direct cost;
8. reproducibility of the results in the same conditions;
9. repositories and documentation of available services.



1. Scientific libraries & domain specific tools

1. **IaaS** services were not designed to target sci. community:
 - spend some time to install on the VMs that are acquired the specific libraries and tools (usually sys admin)
 - licences problems
 - + control of the VM similar to the desktop
2. **PaaS** level: the control is lost,
 - the deployment is depending on the services available
 - the majority serving are web applications, not scientific ones
3. **SaaS** level
 - only few specialized services for sci computations can be use
 - + R for statistics



2. Programming languages

1. The **IaaS** offer is friendlier with the scientist,
 - + allow to activate or install on VMs that are acquired the preferred programming language execution environment
 - usually the code is designed and prepared on the desktop or local cluster, not on the acquired resource, as being costly to spend time in the design on the Cloud.
 - Compared with a Cluster or a Grid, the Cloud
 - + more open in what concerns the permissions of the environment settings
 - while the communications between different resources is more restrictive
2. The **PaaS** is less friendly
 - the common languages are Python and Java more appropriated for Web applications than the scientific ones
3. **SaaS**
 - do not offer a possibility to develop/deploy own code.



3. Store large data

1. IaaS

- + Storage-aaS, Data-aaS, Database-aaS, Backup-aaS ...
- + Storing data in the Cloud a practice for any citizen
- + Unlimited capacity is perceived by the user
- Usage is currently prohibited by the costs of the data transfers and the unconventional forms of stores in Clouds.
- + *the Map-Reduce mechanism : potential to impact the way in which data science is done at this moment.*
 - + data hosting and processing where is stored
 - + storage capacity augmented with algorithmic support for data processing

2. PaaS

- + support for processing the data and the protection of data through the embedded fault tolerance mechanisms

3. SaaS

- + several services dealing with large data
(e.g. image collections, videos, e-mail boxes)



3. Sharing and securing access

- Sharing data, results, codes and basic information is a challenge for each research group.
 - A common virtual place for the shared items inside a institution
 - In the case of the teams working in multiple locations, the sharing becomes a complex problem.

1. IaaS

- + 'permanent' available resources

2. PaaS

- + control access rights

3. SaaS

- + collaborative tools (e.g. virtual spaces for video conference)

Problems (-) to all levels: security leaks



4. Easy deployment of new codes

1. IaaS

- a difficult task similar with the Cluster case and less complicated than in Grids

2. PaaS

- + if the application is complying with the programming style, the deployment is done smoothly;
- difficult to match requirements like response of the remote code under a certain very short period (as synchronous approach is often used in the concept implementation).

3. SaaS

- + the scientist who is able to deploy a new code becomes a Cloud service provider.

5. HPC

- Cloud services are not built to satisfy such HPC requests
 - The resources that are available are usually standard ones.
 - Only recently several providers have started to offer special services to the scientific community, like *clusters-on-demand*.
- Parallel codes
 - Cannot run any Cloud
 - Messaging is costly
- Considerable costs of using a large no.resources for long time
 - similar costs are encountered when using the Cluster or the Grid, but they are hidden in the indirect costs (not accountable per appl) of the institutions that are offering the resources.
- At PaaS and SaaS level
 - no offer for parallel computing or special hardware.



6. Reproducibility of the results

- Scientific applications should be independent from the execution environments: their performance measures should not be environment dependent.
 - + In Clusters are usually offered homogenous resources,
 - in Grids and Clouds the execution environments can vary.
- Control of QoS is essential in this context to all deployment levels.
 - Currently the SLA mechanisms are focusing more on provider requirements than the scientist ones.

7. Repositories and documentation of services

- + The Cloud services are usually well documented by the providers as being in their interest for take-ups
- Diversity in terms of interfaces and the lack of standards.
 - at IaaS there are several tries to standardize the interfaces and to make repositories of services
 - at PaaS and SaaS such initiatives are missing

Analysis of 1-7:

=> Explain why most of the reports that are concerning public Clouds usage for scientific applications are referring to IaaS

=> Not all Clouds are ready to serve the sci applications

Correct usage of Clouds?

- Main characteristics that differentiate the Cloud from the earlier distributed computing paradigms:
 1. Elasticity
 2. Pay-as-you-go concepts
- Few scientific applications are able to be elastic in terms of the requested resources
 - Increasing and decreasing the number of resources depending to the inputs is an atypical behavior for a scientific application.
- Examples:
 - Classification of gene expression datasets and brain imaging workflow using Aneka.
 - Bioinformatics applications based on BLAST using Azure
 - Biomolecular simulations and astronomical image mosaicking were case studies for using SAGA and Amazon.

Scientific Clouds?

- National or international Scientific Cloud
 - under discussion by many research agencies.
- Ideas:
 - Follow the basic concepts of the Grid initiative of sharing resources between the infrastructure providers
 - Augment such services with the ones offered by Cloud technologies, like virtualization or multi-tenancy,
 - Involve also commercial Cloud providers. Pro and contra
- European initiative:
 - Helix-Nebula (www.helix-nebula.eu)

National Scientific Cloud?

- A national Scientific Cloud can
 - allow the access of a larger scientific community,
 - ensure a better customization according to the user needs of the execution environments,
 - implementation of energy efficiency procedures,
 - a better control of the costs of the resource consuming,
 - bursting versus commercial Clouds in cases of peaks.
- Contras
 - cost model that is revealing the usage of e-infrastructure as a direct cost of the research activity,
 - potential leakage of critical data.

PaaS support for scientific applications

- PaaS offer is currently limited in what concerns the scientific application needs
 - the design concepts of most of the PaaS that are targeting long-running Web applications with an unpredictable number of users.
- + Research activities involving large data available on the Internet are partially matching this target
 - e.g. related to network research, social behavior, data mining

Requirement for PaaS to support scientific apps

The scientist usually can access a local environment allowing to perform early experiments & develop the codes

⇒ the Hybrid Cloud paradigm is more appropriate

⇒ the codes are developed locally, on a Personal or Private Cloud

⇒ when more resources are needed they are ported to a Public Cloud.

The portability of the codes between Clouds is in this context an issue for the scientist.

Two types of Cloud platforms

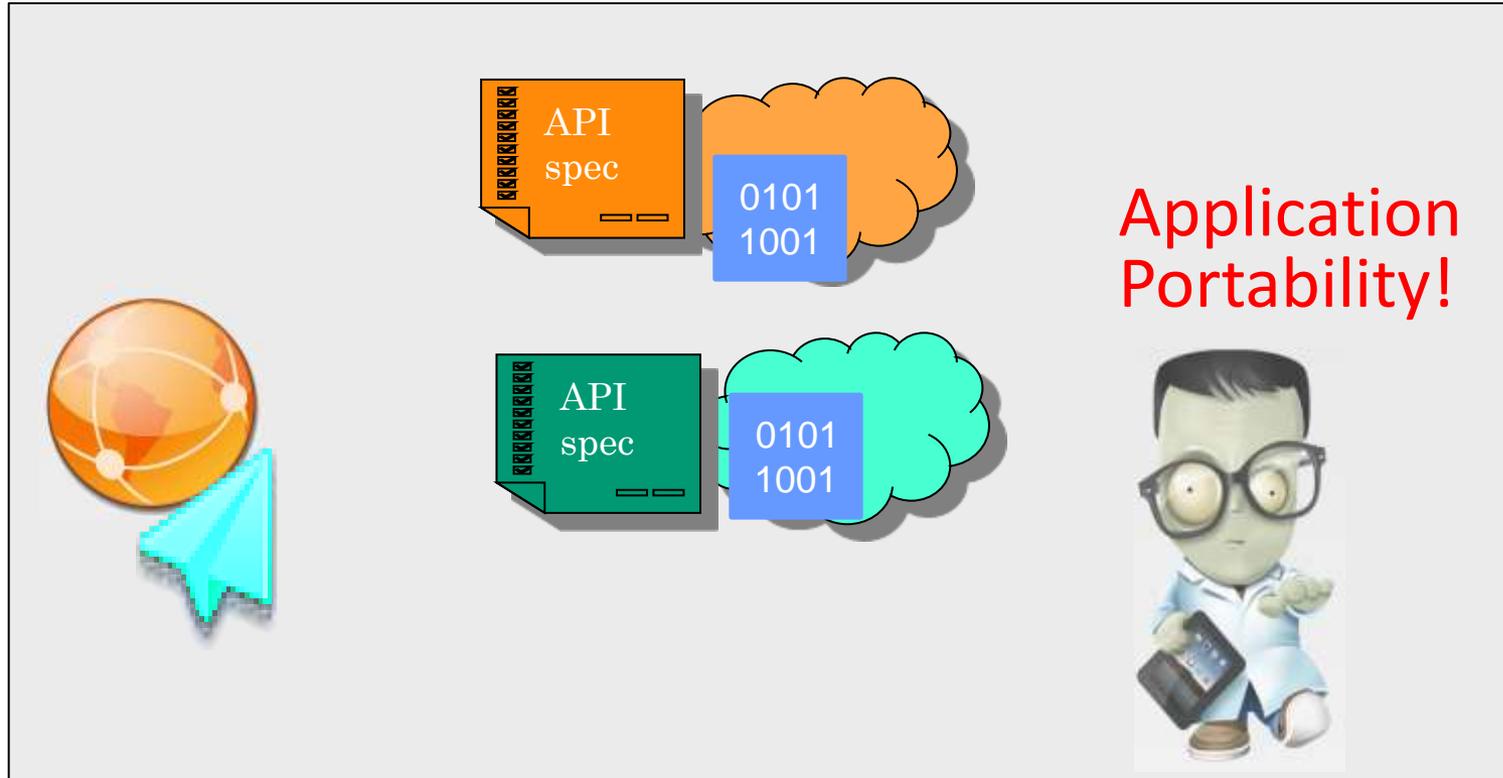
Platform Service | Hosting | Integrated solution

- ❖ Well known examples:
 - Google's AppEngine, Microsoft's Azure, RackSpace's CloudSites, Amazon's Beanstalk, Salesforce's Force.com, Joyent's SmartPlatform
- ❖ **Open-source** only to develop apps
 - to allow customization

Platform Software | Software service | Cloudware Deploy-based solution

- ❖ Deployment of middleware in data centers
- ❖ Easy way to deal with portability and interoperability (framework category)
- ❖ **Open-source** have the potential to impact the market as...
 - PVM/Parallel
 - Globus/Grid

mOSAIC's marketing motto: "Flying through the Clouds"



**mOSAIC: Open-source API
and Platform for multiple Clouds**



mOSAIC as R&D collaboration effort



www.mosaic-cloud.eu

Consortium:

1. Second University of Naples, Italy
2. Institute e-Austria Timisoara, Romania
3. European Space Agency, France
4. Terradue SRL, Italy
5. AITIA International Informatics, Hungary
6. Tecnalia, Spain
7. Xlab, Slovenia
8. University of Ljubljana, Slovenia
9. Brno University of Technology, Czech Republ.



September 2011: 1st API implement. (Java)

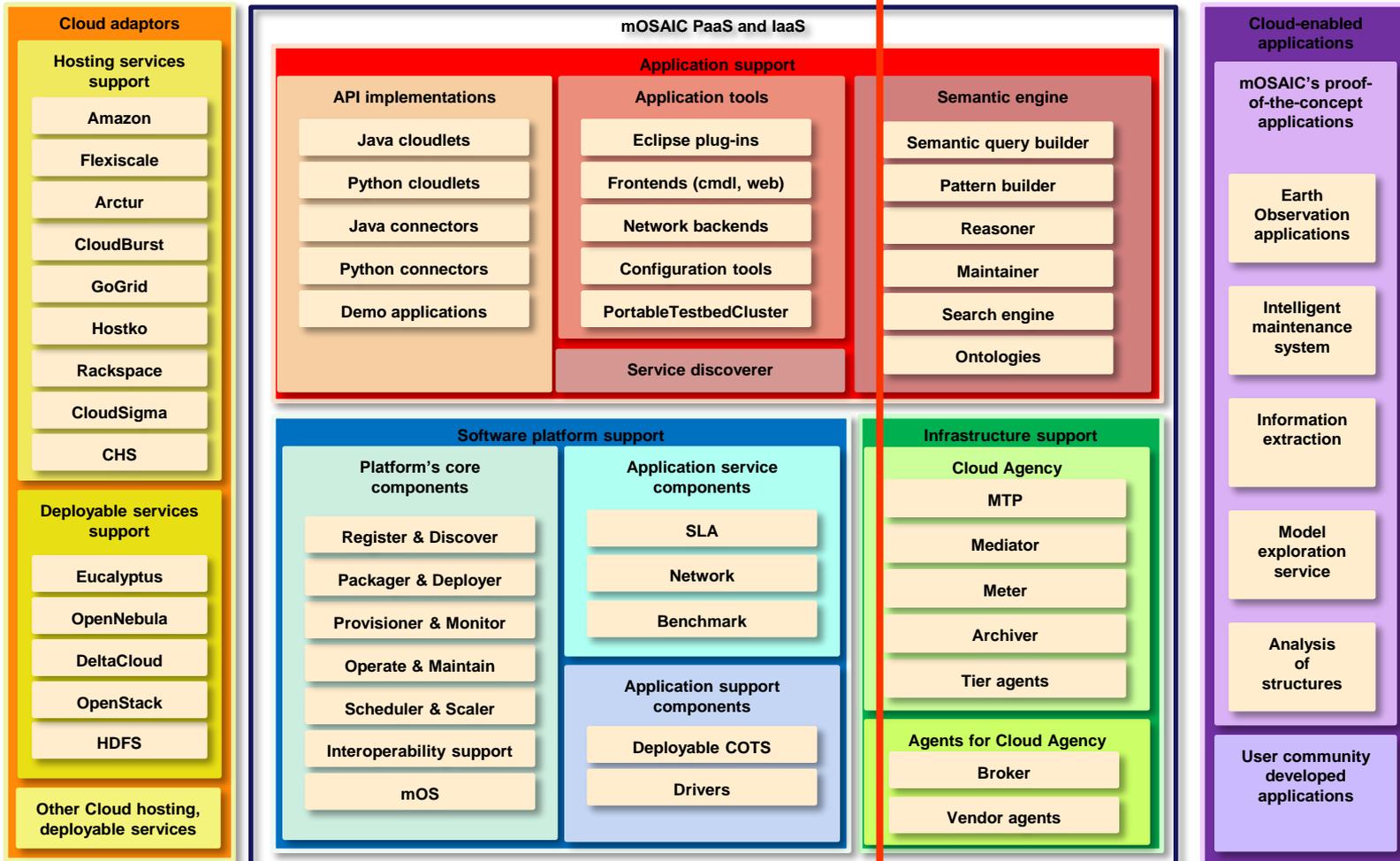
September 2012: 1st stable PaaS,
2nd API impl. (Python)

March 2013: Full software package



mOSAIC's layered architecture

Open-source and deployable PaaS



OS repository: <https://bitbucket.org/mosaic>



How to use it? Scenario:

- ❖ **Write component-based application**
 - Languages: Java, Python, Node.js, Erlang
 - Communications through message passing
 - Respect the event-driven style of programming
 - Find the proper functionalities with the Semantic Engine
- ❖ **Debug your application on the desktop or on-premise server(s)**
 - Within Eclipse
 - Use Personal Testbed Cluster using VirtualBox for the VMs
- ❖ **Deploy your application in a Cloud**
 - Assisted by Cloud Agency and Broker (with SLAs)
- ❖ **Monitor & modify the applications**
 - Control the life-cycle of the components (start/stop/replace)

VIDEO DEMOS on YouTube – keywords “mOSAIC Cloud Computing”



Open-source Platform Software

Product	AppScale	Cloud Foundry	ConPaaS	mOSAIC	OpenShift	TyphoonAE	WaveMaker
Owner	Univ. California	VMWare	Contrail Consortia	mOSAIC Consortia	RedHat	Tobias Rodäbel	VMWare
Site	appscale.cs.ucsb.edu	www.cloudfoundry.com	www.conpaas.eu	www.mosaic-cloud.eu	open shift.com	code.google.com/p/typhoonae	www.wave maker.com
Repository	appscale.googlecode.com/svn/	github.com/cloudfoundry	www.conpaas.eu/download/	bitbucket.org/mosaic	github.com/openshift	code.google.com/p/typhoonae/downloads/	dev.wavemaker.com/wiki/bin/
State	1.5/Jul 2011	0.x , Beta	0.1/Sep 2011	0.5/Jun'12, Beta	Production	0.2/Dec 2010/beta	6.4.4/Dec 2011
Languages	Python, Java, Go	Java, Ruby, Node.js , Groovy	PHP	Java, Python, Erlang, Node.js	Java, Python, Perl, PHP, Ruby	Python	Java
Data Support	HBase, Redis Hypertable, MySQL Cluster, Cassandra, Voldermort, MongoDB, Memcache-DB	MongoDB, SQLFire, PotsgreSQL, Redis	Scalaris, MySQL, XtreamFS	Riak, CouchDB, Mem-cacheDB, Redis, MySQL	MySQL, MongoDB, Amazon RDS	MongoDB, MySQL, Berkeley DB JE	Amazon S3, Rackspace
OS	Ubuntu, CentOS on Xen, KVM	VMWare image	XtreemOS image	mOS image, Linux	Red Hat Virtualization	Debian, Ubuntu	VMWare image
Messaging	Channel	RabbitMQ	Own design	RabbitMQ	Own design	RabbitMQ, ejabberd, Channel	Own design
Clouds tested	Amazon EC2, Eucalyptus	VMWare	Own testbed	Amazon EC2, Eucalyptus, OpenNebula, Flexiscale ...	RightScale Rackspace, Smart-Cloud, Amazon	Google	EC2, Rackspace, OpSource, Eucalyptus
Interface	CLI, Web	CLI	Web	CLI, Web, REST	CLI, REST	CLI	Studio

Conclusions:

- Identified the pro and contras using the Clouds for sci appls
- Help to develop scientific applications without the need of tedious installation can be assured by Cloudwares
- The development of proof-of-the-concept scientific applications of mOSAIC have prove the above