CLOUD INTERCHANGEABILITY
Redefining Expectations

André Monteiro, Joaquim Sousa Pinto
Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro, Campus Santiago, Aveiro, Portugal

Cláudio Teixeira, Tiago Batista
Instituto de Engenharia Electrónica e Telemática de Aveiro, Universidade de Aveiro, Campus Santiago, Aveiro, Portugal

Keywords: Cloud computing, Systems interoperability, Interchangeability, Standardization, Network resources.

Abstract: The large growth of the last few years in cloud computing has been overwhelming, either on resources, deployment or complexity. Still, it lacks a sustaining basis such as standard methods and deployment settings to potentiate a full evolution. Interoperability and interchangeability are expected to overcome vendor lock-ins. This paper overviews the problem with a state-of-the-art review and reflects some of the undergoing approaches.

1 INTRODUCTION

The emerging of the Cloud era was a globally disperse evolution, scattered by different hardware and technologies but also in infrastructure and composing layers. Early attempts for standardization were rather shy because cloud computing originated in the private sector (Caryer et al. 2009). As a scattered development, with few defined standards, each provider developed its cloud infrastructure on their own, instead of using normalized settings, a very similar situation to the appearance of the TCP. This vision from within held back a bit on standardization (Grossman, 2009).

Standards are important because they ensure interoperability in a well-designed formalism (Zeller et al., 2009). They define a persistent basis to allow sustained development, even though it can be a bigger start up obstacle. Thus, developing under standards can be more costly and harder at the beginning of the process, but on the long term it compensates in benefits formalism (Zeller et al., 2009). However, it is widely held that common best practices and standards will be needed to realize many of the benefits being touted for cloud computing (Lee, 2010).

Nonetheless, cloud computing usage went sky high because of its major functionalities: on-demand use, ubiquitous access, pay-as-you go service, resource saving and elasticity. As an obvious result, the convergence of the whole Cloud eco-system turns into a colossal task, restricted in edgeless connectors and in the business lobbies that cause a dreadful inertia.

Interchangeability, more than interoperability, is the path to follow. By having a common representation for operationally important artefacts, practitioners have some flexibility to move models from one environment to another formalism (Zeller et al., 2009). There are some proposals of evaluation techniques to determine the best cloud service (Chan and Chieu, 2010), but is avoidance the way? We change Internet and TV provider as we want, without changing laptops or TV sets. Why shouldn't applications be moved across providers when we decide to change the service supplier? No doubt that with the standardization Cloud Computing would evolve to its maximum potential and would give users full throttle to their applications. Fulfilling the cross-cloud paradigm is a hard assignment but it surely would free the users to the cloud sky.

2 CLOUD PROVIDERS

One of the major problems is to transpose a common

Monteiro A., Sousa Pinto J., Teixeira C. and Batista T..
CLOUD INTERCHANGEABILITY - Redefining Expectations.
DOI: 10.5220/0003393301800183
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 180-183
ISBN: 978-989-8425-52-2

application to the cloud structure. As the operating system's incompatibility wasn't already enough, the crossing between cloud providers gets even more difficult. The plot gets deeper when analyzing some of the providers. However, in order not to lose any share of the market and the train of edge technology, some providers are going both ways. These few are moving towards interoperability and open standards, applying their proposals to task forces to prevail as a standard. If they succeed, it should give them a head start to boost their offered services and blow away the competition outside the taskforces.

2.1 API Overview

An API is by definition the interface where software can communicate with the infrastructure. As result of the vendor's lock-in, the software attached to some specific API loses some advantages, like reusability. By analyzing some of the vendor's API's, some of the issues tend to be noticed even further. For example, naming for operations is not univocal and even in similar methods input and outputs are quite separate. As an exemplificative instance rather not an exhaustive documentation finding, only three IaaS providers and a small group of operations were chosen. Table 1, 2 and 3 show the results of elicitation.

2.2 Amazon EC2 API

Amazon has been one of the greatest propellers for cloud computing with the launching of a limited public beta of EC2 on mid-2006 (Barr, 2006). In fact, its API has become a *de facto* standard (Grossman, 2009), being used by cloud-based applications like Eucalyptus and Nimbus. This commitment can be overlooked as an advantage but it can also take out and limit developing control on the future roadmap.

Table 1: Amazon API example (AWS, 2010).

	Create Image	List images	Run instance
Name	CreateImage	DescribeImages	RunInstances
Input	InstanceId*, Name*, Description, NoReboot	ExecutableBy.n, ImageId.n,Owner.n, Filter.n.Name, Filter.n.Value.m	ImageId*, MinCoun*t, MaxCount*, […+21]
Output	requested, imageId	requested, imagesSet	*required

2.3 RackSpace Cloud API

The RackSpace Cloud has been operational since its creation in 2006. The Cloud Servers API is implemented using a RESTful web service interface. Likewise in the Rackspace range, Cloud Servers share a common token authentication system that allows seamless access between products and services (RackSpace, 2010).

Table 2: RackSpace API example (RackSpace, 2010).

	Create Image	List images	Run instance
Name	image	images	servers
Input	serverId, name		name, imageId, flavorId, metadata, personality
Output	id, serverId, name, created, status, progress	id, name, updated, created, status	id, imageIdm hostId, progress, adminPass, metadata, addresses

2.4 Sun Cloud API

The Sun Cloud is currently on beta stage at Project Kenai. The documentation describes an HTTP-based RESTful API for managing server-side objects (Sun, 2010).

Table 3: Sun Cloud API example (Sun, 2010).

	Create Image	List images	Run instance
Name	Create VM	Get Cluster	Control VM start-VM
Input	name*, description, from_template, data_disk		
Output	op, progress, target_uri, status_uri, status	name, uri, tags, vms, controllers	*required

3 CLOUD STANDARDIZATION

To address these communication problems between clouds, several organizations joined in combined efforts to deliver cloud standards.

The European Telecommunications Standards Institute (ETSI) has started a project for standardizing the use of grid and cloud computing technology in the context of telecommunication, with the formation of the Technical Committee GRID (Rings et al., 2010). These include analysis of grid and cloud interoperability gaps and surveys, comparisons between grid, cloud and telecommunication systems among others related to grid and Next Generation Networks.

The IEEE society also created a study group for cloud standards, with a call for participation open to all interested. IEEE believes that major impediment to the growth of cloud computing is the lack of comprehensive high-level portability and interoperability standards (IEEE, 2010).

Recently, the Distributed Management Task Force (DMTF) assumed the need for open management standards for cloud computing (DMTF, 2009). DMTF focus its goal in developing cloud resource management protocols, packaging formats and security mechanisms to facilitate interoperability and portability between compute clouds.

Already set is the open standard for cloud images Open Virtualization Format (OVF), released by DMTF and almost in a final version. It defines a portable and extensible format for software to be run in virtual machines.

4 UNDERGOING PROJECTS

Whereas providers continue to enclose their specifications in a dark cloud, the key to interchangeability rests on middleware, API's and also OS level. Some may consider it as an evasion path, but it can offer the services of several providers, as long as cloud services' API's are tested and included. Some research projects are being made currently, with focus in simplifying and unifying the management of applications under a cloud infrastructure and thus provide interchangeability. Interchangeability, or portability between clouds, remains postponed.

4.1 Red Hat DeltaCloud

Red Hat announced in mid-2010 the DeltaCloud project, defending that third-party governance was necessary to achieve true interoperability and portability (McGee, 2010). DeltaCloud is developing an API to deal with several clouds service providers and publicizes any-platform access REST API and backward compatibility across versions, providing long-term stability for scripts, tools and applications (Red Hat, 2010). The project is running and accepts contributions as it is held in Apache Incubator. With a full implementation it should be possible to start and run instances on a private cloud and repeat the operation with public clouds seamlessly.

4.2 IBM AltoCumulus

As a response to the cloud heterogeneity, IBM started an effort group to develop a middleware platform called AltoCumulus (IBM, 2010). The project, started in mid-2009, aims to provide a uniform, service oriented interface to deploy and manage applications in various clouds and also provides facilities to migrate instances across clouds using repeatable best practice patterns (Maximilien et al., 2009). This approach tends to be very comprehensive since it intends to provide interoperability for different cloud layers as IaaS, PaaS and private clouds. As the previous projects, its API lies on REST support.

4.3 Unified Cloud Interface Project

As DeltaCloud, the UCI sets to develop a standardized cloud interface for the unification of various clouds' API's (CCIF, 2010). The goal is to include the entire infrastructure stack as well as emerging cloud centric technologies through the merged API. This is done by describing a semantic cloud data model with Resource Description Framework (RDF) and therefore using it to control these web resources.

4.4 Open Cloud Computing Interface

The OCCI working group is developing a clean, open API for Infrastructure as a Service (IaaS) based Clouds (Sun, 2009). The API is intended interface all Cloud providers, private or public, and to ensure portability, interoperability and integration between Cloud service providers and end-users. This is being proposed as an extendable RESTful based API.

4.5 InterCloud

The InterCloud is defined as being a global cloud interconnecting other clouds within the network. Identifying a profile of protocols and formats is one part of the interoperability puzzle (Bernstein et al., 2009), where peer clouds must dialog. A key concern must be the assurance that if one component fails the whole system doesn't collapse.

4.6 Operation System Layer

The Distributed Computing Research Group at the University of Newcastle proposes a byte-code style layer (Wallis et al., 2010). Their insight is that this kind of layer will allow the code execution transversely across multiple hardware platforms. The project is currently running and specifies cross-platform execution, component swap communication architecture, associated security, among others.

5 CONCLUSIONS

Cloud computing has become a phenomenon today, amongst every single kind of user and provider. Nevertheless, its unfocused development lead to a enormous heterogeneity and islets with low interoperability and rare interchangeability. Yet, the perspective of a cloud managed by any service provider or enterprise interoperating is impressive. Application portability is even more formidable. This paper reflects a general overview of the state-of-the-art to show the big path yet to walk.

As standardization and projects converge to a common point, one can only expect greater achievements.

REFERENCES

Caryer, G., Rings, T., Gallop, J., Schulz, S., Grabowski, J., Stokes-Rees, I. and Kovacikova, T. 2009. Grid/cloud computing interoperability, standardization and the NGN. *In Intelligence in NGN ICIN 13th International Conference on*.

Grossman, R.L. 2009. The Case for Cloud Computing. *IT Professional, vol.11, no.2, pp.23-27*.

Zeller, M., Grossman, R., Lingenfelder, C., Berthold, M.R., Marcade, E., Pechter, R., Hoskins, M., Thompson, W. and Holada, R. 2009. Open standards and cloud computing: KDD-2009 panel report. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09)*, New York, USA, 11-18.

Lee, C. 2010. A perspective on scientific cloud computing. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10)*. New York, USA, 451-459.

DMTF. DMTF to Develop Standards for Managing a Cloud Computing Environment, http://dmtf.org/news/pr/2009/4/dmtf-develop-standards-managing-cloud-computing-environment, accessed on December 2010

Chan, H., Chieu, T. 2010. Ranking and Mapping of Applications to Cloud Computing Services by SVD. In *Network Operations and Management Symposium Workshops 2010*. IEEE. Osaka, Japan, 362-369.

Barr, J. Amazon EC2 Beta. http://aws.typepad.com/aws/2006/08/amazon_ec2_beta.html, accessed on December 2010

Amazon Web Services. Amazon Elastic Compute Cloud User Guide. http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide, accessed on December 2010

RackSpace Cloud, Inc. Cloud Servers Developer Guide. http://docs.rackspacecloud.com/servers/api/cs-devguide-latest.pdf, accessed on December 2010

Sun Microsystems, Inc. The Sun Cloud API. http://kenai.com/projects/suncloudapis/pages/Home, accessed on December 2010

Rings, T., Grabowski, J. and Schulz, S. 2010. On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures. *2010 Second Advances in System Testing and Validation Lifecycle,* International Conference on, pp. 99-107.

IEEE Computer Society. IEEE Cloud Computing Standards Study Group. http:/computer.org/portal/web/sab/cloud, accessed on December 2010

DMTF. DMTF to Develop Standards for Managing a Cloud Computing Environment. http://dmtf.org/news/pr/2009/4/dmtf-develop-standards-managing-cloud-computing-environment, accessed on December 2010

McGee, T. Red Hat Deltacloud Aspires to Become an Industry Standard. http://www.linux-magazine.com/Online/News/Red-Hat-Deltacloud-Aspires-to-Become-an-Industry-Standard?category=13398, accessed on September 2010

Red Hat. DeltaCloud, http://deltacloud.org, accessed on November 2010

IBM. Almaden Services Research Project Altocumulus. http://www.almaden.ibm.com/asr/projects/cloud, accessed on September 2010

Maximilien M., Ranabahu, A., Engehausen, R. and Anderson, L.. 2009. IBM altocumulus: a cross-cloud middleware and platform. In *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA '09)*, New York, USA, pp. 805-806.

Cloud Computing Interoperability Forum. Unified Cloud Interface Project. http://code.google.com/p/unifiedcloud, accessed on December 2010

Sun Microsystems. 2009. What is OCCI? http://blogs.sun.com/inthecloud s/entry/what_is_occi, accessed on December 2010

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S. and Morrow, M. 2009. Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability. *4th International Conference on Internet and Web Applications and Services.* IEEE, pp. 328–336

Wallis, M., Henskens, F., Hannaford, M. 2010. Expanding the Cloud: A Component-Based Architecture to Application Deployment on the Internet. In *Cluster, Cloud and Grid Computing (CCGrid)*, 10th IEEE/ACM International Conference, pp.569-570.