Open Source Cloud Computing: Characteristics and an Overview

Naylor G. Bachiega[1], Henrique P. Martins[1], Roberta Spolon[1], Marcos A. Cavenaghi[1], Renata S. Lobato[2], Aleardo Manacero[2]
[1]Computer Science Dept., Paulista State University – UNESP, Bauru, Brazil
[2]Computer Science and Statistics Dept., Paulista State University – UNESP, São José do Rio Preto, Brazil

Abstract - *In an attempt to reduce costs by taking advantage of efficient computing resources, new developed technologies and architectures are gaining wide acceptance in the market. One of such technologies is cloud computing, which uses existing technologies, such as virtualization, trying to solve problems like energy consumption and space allocation in data centers or large companies. This paper presents a study on cloud computing, describing their main characteristics, models of deployment, services, and architectures, including an analysis over its benefits, risks and challenges. It also presents a study over some open-source cloud managers, presenting its advantages and drawbacks. All of this is presented aiming to provide a clear guide for those that are evaluating the possible adoption of cloud technology for their IT problems.*

Keywords: Cloud Computing, Architecture, Virtualization.

Introduction

With the constant growth in the use of computers problems such as power consumption and storage space for data centers are becoming a commonplace. Several solutions have been launched to solve this problem, including Cloud Computing, as named from IBM in 2007 [1].

Cloud computing is devised as a strong trend nowadays, with most of the organizations using or planning to use it. The advantages brought by cloud computing include the reduction in hardware's acquisition and maintenance cost, accessibility, flexibility, and a highly automated process for software upgrades [2].

The cloud can be defined as a network infrastructure based in the share of computing resources along the Internet. The major differential is that clouds try to make the infrastructure's complexity transparent for users. This is performed through the offering of “services” that deliver clients' requests using the Internet. Such offering is enabled by the use of virtualization technologies throughout datacenter’s infrastructure, storing and processing users data outside their local resources [3].

Although the rise in its application, cloud computing is still evolving. Open issues include how clients pay for resources and what resources have to be paid for. Also, there are open problems in its maintenance cost, accessibility, and flexibility.

In this paper we provide a concise review of the main concepts involved with clouds, including virtualization, virtual machines and hypervisors. We also provide a description of some characteristics that must or should be present in clouds. We finish with an evaluation of three open-source cloud managers.

Virtualization

Virtualization is an important concept for the cloud architecture. Most of modern processors support native virtualization, with several solutions implementing virtual machines present in the market. This is useful since it is rare to find dedicated servers using most of its processing capacity. Therefore it is costly to maintain several physical hosts with a different operating system in each one, making desirable and viable to have several virtual machines running in a single physical one.

Virtualized environments are designed through the insertion of a virtualization layer between the hardware and the virtual machine (VM), as shown in Figure 1. A VM enable a more efficient use of hardware resources while executing the user's applications running in a given operating system [4].

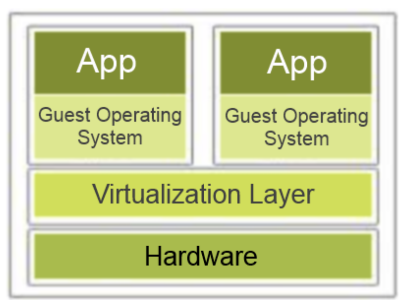

Figure 1. Architecture of hardware virtualization [4].

A. Virtualization Environment

The virtualization layer is the software responsible by the hosting and management of all virtual machines through a VM monitor (VMM). It is a hypervisor running directly over the hardware, with different capabilities for each different architecture. Each VMM executing in the hypervisor implements a hardware's abstraction of the virtual machine and runs a guest operating system. All VM share CPU, memory, and peripherals in order to successfully build a virtualized environment [4].

Unfortunately some hardware instructions cannot be effectively virtualized, since they have different semantics when executed in non-privileged modes. The approaches to circumvent such problems are:

Total virtualization with binary translation: any x86-based operating system can be virtualized by a combination of binary translation and straight execution techniques. This approach translates the kernel to replace non-virtualizable instructions with macros that have the intended effect in the virtual hardware [4].

Assisted Virtualization or Paravirtualization: modifies the guest OS kernel in order to replace non-virtualizable instructions with hypercalls that directly communicate with the hypervisor layer. The hypervisor also provides interfaces (named hypercalls) to other kernel's critical operations, such as memory management, interrupt management, or time management. Paravirtualization differs from total virtualization in the sense that here the unmodified OS does not know that it is being virtualized and sensitive system calls are captured through binary conversion [4].

Hardware Assisted Virtualization: hardware providers are rapidly adopting virtualization, improving the resources to make it easier. Improvements here include Intel's Virtualization Technology (VT-x) and AMD's AMD-V. In both cases the technique is to provide new modes to execute privileged instructions [4].

B. Hypervisor

The hypervisor is a software layer between hardware and operating system, controlling the access of the guest OS to the hardware's resources. In order to work correctly, the hypervisor needs to have control over the real system's resources. It also needs to satisfy certain constraints, such as to provide an exact copy of the real execution environment to the applications running in the virtual machine. There are several hypervisors available, including Xen, VMware, KVM and QEMU [4].

CLOUD COMPUTING

A cloud is a system using concepts as virtualization, emulation, OVF (VM configuration patterns), and Libvirt (an API to manage guest operating systems), and to assemble a new architecture from them. This new architecture basic idea is to provide computing power from demand, where more virtual machines can be added to the cloud when an user needs it. Actual machines can be added in order to increase the computing power and/or storage capacity.

Since hardware virtualization allows the creation of multiple virtual machines over a real machine, cloud computing uses it to create an environment (the cloud), allocating instances (guest operating systems) accordingly to the available resources (physical hosts). These virtual machine instances are allocated to the convenient actual hosts that compose the computing cloud [5].

The key to achieve cloud computing is the “cloud”, which is a massive network of hosts (servers or simple machines) connected as a grid. These computers may work independently or in parallel, when resources are combined in order to create a high performance system [6].

As shown in Figure 2, individual users access the cloud from their own computers, or portable devices, using the internet. For these individual users, the cloud can be seen as a single application to share documents or devices. Both hardware and operating system are made transparent for the user, simplifying the cloud's access by the user.

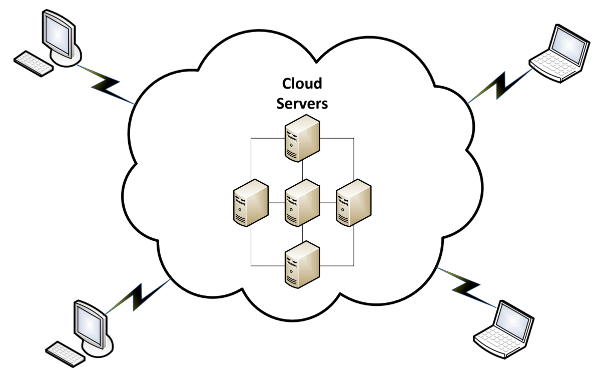

Figure 2 – A typical, simplified, cloud system [7].

A. Classification schemes

According to Sasikala [8], computing clouds can be classified accordingly to the type of users sharing/providing their services/resources. The defined classes are:

Public cloud: the infrastructure is provided to many clients and managed by a third party, that is paid by its usage. In this class several companies may be involved in the same environment.

Private cloud: the infrastructure is built aiming specific clients inside one organization. It can be managed by the own organization or by a third party. In this case, virtualization can be made using proprietary tools.

Community cloud: the infrastructure is shared among several organizations, usually with a common goal. It can be managed by the organizations or by a single service provider.

Hybrid cloud: is composed by two or more of the previous models, demanding transparency in all transfers between them.

B. Architecture

According to Zhang, Cheng and Boutaba [9], the architecture of a cloud computing environment can be divided in four layers: hardware/datacenter, infrastructure, platform and application layers, as shown in Figure 3. The description of these layers follows:

Hardware layer: it is responsible of managing the cloud's physical resources, including servers, network devices, electrical power, and cooling systems. It is usually implemented through datacenters, with hundreds or thousands of servers in order to deal with issues such as fault tolerance and traffic routing.

Infrastructure layer: it is also known as virtualization layer and creates a pool of resources for storage and processing, allocating physical resources through hypervisors such as Xen, KVM or VMware. This layer is essential in the cloud environment since dynamic resource allocation and other important features are provided by hypervisors.

Platform layer: it is built over the infrastructure layer, consisting of the operating systems and frameworks for software applications. It is designed aiming the reduction in the cost of developing applications directly for VM stubs. As an example, the Google App Engine operates in this layer in order to provide a support API for the development of databases, storage and typical businesses rules for web applications.

Application layer: in the top level of this hierarchy, the application layer is composed by actual cloud applications. These applications differ from conventional ones since they can take advantage from automatic resizing in the resources in order to achieve a better performance, availability and operational cost.

This layered architecture resembles the OSI reference model for computer networks, providing modularity, flexibility and independence from changes in each layer. This allows the reduction in costs of management and maintenance, at the same time it can execute a wider range of applications.

In order to achieve such characteristics it is expected that a cloud computing environment would offer the following properties [10], also appearing in Figure 3:

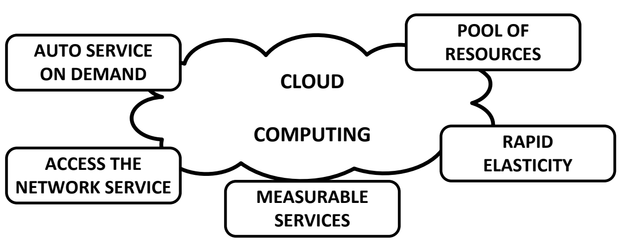

Figure 3 – Cloud computing properties [11].

Automatic service on demand: computational services are provided automatically, without human intervention over the service provider;

Wide access to network services: since computing resources are available through the internet, they must be easily accessible via standard protocols by any kind of device (mobile, handheld, or desktops);

Pool of Resources: provided computing resources (physical or virtual) have to serve multiple users, being allocated and reallocated accordingly to the demand;

Rapid Elasticity: services must be fast and made available whenever necessary. Users of them must feel as they have unlimited resources, which can be acquired in any amount, anytime. The elasticity property appears in three components: linear scalability, use by demand, and payment of what is consumed;

Measurable Services: the management systems used by the cloud must control and monitor each resource, automatically, for each kind of service (storage, processing, and bandwidth). This monitoring must be transparent for both entities involved (service provider and user).

Zhang, Cheng and Boutaba [9] establish that cloud computing provides services in a form that largely differs from the form provided by conventional computing datacenters. The differences include:

Multiple tenants: in a cloud environment services owned by multiple providers can be located in a single data center. With this approach the issues with performance and management of these services can be dealt by all service and the infrastructure providers. The layered architecture offers a natural division of duties, that is, the owner of each layer will have to deal only with the specific goals associated with that layer. Unfortunately, the multiple tenancies also create problems in the understanding and management of the interactions between parts.

Sharing of the pool of resources: the infrastructure provider offers a pool of resources that can be dynamically allocated to the resource consumers. This capability creates a great flexibility to the cost-effective management of resources. As an example, a service provider can take advantage of the technique of VM migration in order to maximize resource utilization, what implies in the reduction of costs associated with cooling and power consumption.

Access through a worldly distributed network: clouds are usually accessible from the Internet, therefore, any device connected to it, either a cell phone or a desktop, is capable of using the cloud services. Even more, in order to achieve a high performance and availability, many of the current clouds are composed by several datacenters distributed over the world.

Service Oriented: cloud computing adopts a service-oriented operational model, putting a strong emphasis in service management. Each provider offers his service trying to guarantee a Service Level Agreement (SLA), which is negotiated with the users of that service.

Dynamic resource provisioning: differently from conventional systems, where resources are fixed, in clouds we have the capability of dynamically adjust the amount of offered resources by the acquisition and publishing of extra resources by the service provider, guided by current demands.

Self-organization: the property of dynamic resource provisioning implies in the ability of providers and clients to adjust their resources upon demand. Resources can be allocated or returned to the pool depending the current needs. This flexibility results in the elasticity property.

Price-based utilities: cloud computing uses an economical model based in "pay what you use". The exact price of processing may be different for different services. For example, a software provider may rent a VM in a by-hour basis, while other may charge the service by the number of clients served. Although services priced by-use may reduce client's costs, they introduce difficulties in the management of the whole operation.

There are also important differences that distinguish the model of cloud computing from the traditional model of computing. Table I, adapted from [12], summarizes these differences.

TABLE I
DIFFERENCES BETWEEN CONVENTIONAL AND CLOUD MODELS

	Conventional Computing	Cloud Computing
Acquisition Model	Hardware Physical space Infrastructure of installation and operation	Service acquisition
Business Model	Cost and depreciation of assets Administrative overhead (maintenance, support, safety of equipment, refrigeration)	Payment based on demand
Access Model	Internal network Intranet	Internet, through various types of devices (not just desktop computers)
Technical Model	One tenant Without sharing Static	Scalable, Elastic, Dynamic

Additionally it is known that cloud computing and grid computing share some goals, including cost reduction, flexibility and reliability through the use of third party hardware. They differ in the way they allocate resources, where in grids it is attempted a more homogeneous allocation and in a cloud the allocation occurs on demand. Also, the virtualization in cloud computing allows for a greater separation between the resources used by all users.

C. Classes of services

Services in cloud computing have different levels of support, accordingly to what is offered to clients. There are three classes of services, depicted in Figure 4, named IaaS (for infrastructure), SaaS (for software) and PaaS (for platform) [5]. A short description of them follows:

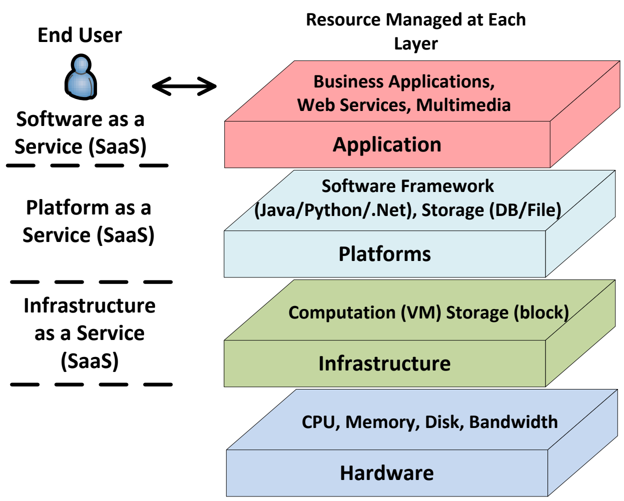

Figure 4 – Cloud computing architecture (Adapted from [9]).

Infrastructure as a Service (IaaS): in this type of service clients are provided with processing, storage, network bandwidth, and other computing resources, being able to reconfigure them as needed. Clients do not manage or control the infrastructure of the remaining cloud, paying only for what is used. Amazon Elastic Compute Cloud (Amazon EC2), Eucalyptus, OpenNebula and OpenStack are examples of providers in this class [8].

Platform as a Service (PaaS): in this class clients get an environment for development, test and deployment of their own applications, disregarding the needs of infrastructure (memory, storage, processors, etc.). Google Apps and Microsoft Azure are examples of services in this class [8].

Software as a Service (SaaS): here the applications are the service provided, with clients demanding the execution of specific programs. The applications can be accessed from several types of devices, usually from a web browser. The client has no control over the infrastructure or even the application [8].

D. Benefits from cloud computing

According to Veras [11], the main benefit brought with the use of cloud computing is scalability. With the resource provisioning provided by the cloud, based on demand, it is easier to scale the system, introducing more resources when they are needed. This allows for reduction in power consumption and management effort, optimizing the use of servers, network and storage space. The economics of clouds involve the following aspects:

Economy of scale from the providers view: it is achieved from big datacenters, minimizing operating costs related to power consumption, personnel, and management. The minimization is a direct result of the assembly of multiple resources in a single domain.

Economy of scale from the demand view: occurs due to the demand aggregation, reducing inefficiencies resulting from load variations, increasing server's usage.

Economy of scale from the multitenancy view: since the degree of sharing can be increased, it is possible to reduce the cost of management of servers.

EVALUATION OF OPEN SOURCE CLOUD MANAGERS

We analyzed three open-source cloud managers, OpenStack, Eucalyptus e OpenNebula, which were chosen because they have better documentation available. In order to perform the evaluation we used six personal computers, configured with 4GB of memory, running 2.66GHz Core 2 Quad processors, linked through a 1 Gbps network and running Ubuntu Server 10.10 64bits. One computer was used as the cluster's manager, while the remaining five were used as computing nodes, using KVM and working as an IaaS model.

A. Eucalyptus manager

As the other managers, Eucalyptus can work in the all-in-one model, where the main services include data storage, VM images server, cluster control and cloud management, and are offered through a single or distributed servers. A single server topology can be seen in Figure 5.

It operates verifying which nodes are part of the cloud and aggregating their resources accordingly to the profiles of the

VM instances present. The cluster uses a round-robin policy to select and execute a VM image in a given node. Once started, that instance can be accessed through its public IP address, attributed by the manager.

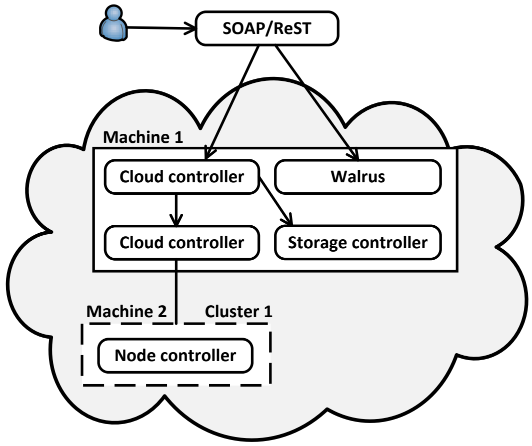

Figure 5 – Eucalyptus topology for a single cluster [13].

As it can be seen from Figure 6, a cluster with 5GB of memory can generate five profiles with different amounts of RAM in each one. They can be selected afterward by the demand needs from each client.

```
Every 2.0s: euca-describe-availability-zones verbose    Sat Jan 14 15:31:48 2012

AVAILABILITYZONE        cluster1         10.10.34.25
AVAILABILITYZONE        |- vm types      free / max   cpu    ram  disk
AVAILABILITYZONE        |- m1.small      0010 / 0010   1     192     8
AVAILABILITYZONE        |- c1.medium     0010 / 0010   1     256     8
AVAILABILITYZONE        |- m1.large      0010 / 0010   1     512    10
AVAILABILITYZONE        |- m1.xlarge     0005 / 0005   1    1024    10
AVAILABILITYZONE        |- c1.xlarge     0000 / 0000   1    2800    10
```

Figure 6 – Profiles of VM instances in Eucalyptus.

B. OpenNebula manager

OpenNebula uses the Ruby language to implement communication among nodes.

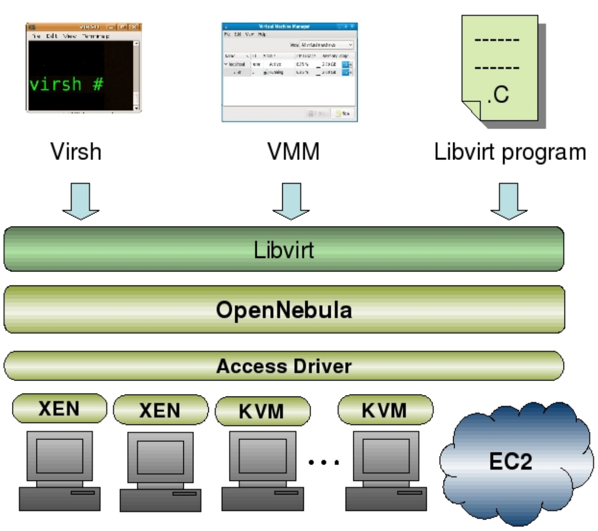

Figure 7 – Libvirt API 1.4 API [14].

It also uses Libvirt (a framework for the creation of VMs), as shown in Figure 7. In OpenNebula the requests for the creation of VM are managed with Libvirt, which translates them to the different hypervisors.

C. OpenStack manager

OpenStack is a collection of open-source projects that can be used, by companies and service providers, to configure and run their storage infrastructure. NASA and Rackspace, among others, contributed massively for its maturation. Rackspace provided a platform for object storage, while NASA contributed with OpenNebula's platform. As it can be seen in Figure 8, there are three main service components in OpenStack:

- Computing node (Nova);
- Storage (Swift);
- Image service (Glance).

The computing node (Nova) is the OpenStack's controller. All activities necessary to support the life cycle of the VM instances are managed on it, using the Libvirt API to interact with the supported hypervisors.

Swift provides a distributed object storage service. It is similar to the Amazon Web Services – Simple Storage Service (S3).

Glance is a search engine and a VM image retrieval system. It can be configured to use any of the storage back-ends available.

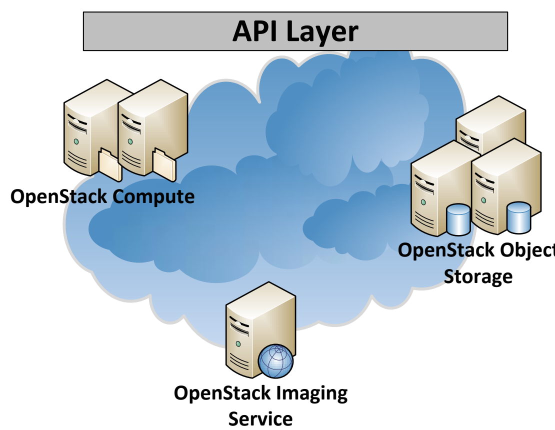

Figure 8 – OpenStack's architecture [15].

OpenStack has also several installation options, being able to be installed in one or more servers, with or without virtualization support. It has to be noted that when there is no virtualization support, it is necessary to emulate it through QEMU.

D. Identified limitations in the managers

With our analysis some flaws, or limitations, were identified in the managers evaluated:

Failover: all tested managers do not have systems against failures, either for the manager or its nodes. For example, if the manager crashes, the remaining nodes can still work normally, but if a node becomes unavailable, that instance is lost, needing a human intervention in order to upload it in another VM.

Scheduler: the choice of which physical node will receive each VM instance is done by the manager's scheduler, what is more effective if dynamic algorithms are in use. Eucalyptus has three algorithms for instance scheduling, round-robin, Greedy (allocates to first found node), and PowerSave (turn off nodes that are not running any VM). OpenNebula and OpenStack have also similar static algorithms. In all cases, the manager is unable to schedule nodes in a dynamic, and more efficient, approach.

Geographical dispersion: for geographically distributed clouds, the manager should be able to reschedule a given VM instance accordingly to the region from which it gets more accesses. However, in many situations, virtualized applications are not ready to be accessed through the network, and do not take location as a parameter in the VM creation.

Power consumption: nodes that are not executing any instance should be turned off or put in a low consumption mode. However, in order to achieve this manager has to have schedulers capable of activating or reactivating nodes as soon the demands requires that. Among the managers evaluated Eucalyptus has the PowerSave algorithm that is capable of this.

Data protection: in all open-source cloud managers there is no a policy or control for data backup. Another issue is that the managers use third party solutions for storage, such as NFS or SAN. These solutions make harder to guarantee data redundancy, especially due to network bandwidth that constrains the speed of data transfers among the nodes. At current systems, this bandwidth should be at least 10Gb/s, in order to be efficient.

In Table II we present a short review of the problems just described. From that it is possible to devise that none of them present solutions for those problems. It is important to note that some of them are managed by proprietary managers.

TABLE II
PROBLEMS WITH OPEN SOURCE CLOUD PROVIDERS

Manager	Storage Virtualization	High Availability	Scheduler	Recovery Data
Eucalyptus	No	No	Static	No
OpenNebula	No	No	Static	No
OpenStack	No	No	Static	No

CONCLUSION

Since cloud computing is becoming more present nowadays, with several companies and organizations adopting this approach, it is very important to understand it. To have this understanding is useful to a better dimensioning of its capabilities, vulnerabilities, and risks. Among the major benefits from clouds there are the minimization of power consumption and physical space, easier provisioning, and easier access to external interfaces (APIs), among others characteristics.

Accordingly to the performed tests, OpenStack presented the best results. This includes a complete documentation, active community, concerns with bug fixes, easy installation, a good quantity of different VM images, allowing assisted instance migration (in the licensed version).

The other two managers presented some deficiencies, such as lack of documentation, obscure architecture, and use of third party solutions in the platform. These problems are an obstacle for a faster adoption of cloud computing by some organizations.

Therefore, before adopting cloud computing it is important to identify which provider offers services that are more relevant to the company or person adopting it. This includes checking for reliability, reputation, accessibility, migration capabilities, and clear contracts, for example.

Finally, we hope to have provided an useful guide for people interested in adopting cloud computing. To do so, we provide a concise description of clouds and the technologies associated to them, as well a brief comparison of the most relevant open-source cloud managers.

REFERENCES

1. HE, Zhonglin; HE, Yuhua. Analysis on the security of cloud computing. Proc. Spie, Qingdao, China, n., p.7752-775204, 2011.
2. BHADAURIA, Rohit; CHAKI, Rituparna. A Survey on Security Issues in Cloud Computing. CORR, P. abs/1109.5388, 2011.
3. SABAHI, F. Cloud computing security threats and responses. Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on May 2011.
4. VMWARE. Understanding Full Virtualization, Paravirtualization, and Hardware Assist. Available at: <http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf>. Last access: 09 July 2012.
5. BUYYA, Rajkumar; BROBERG, James; GOSCISNSKI, Andrzej M. Cloud Computing: Principles and Paradigms. John Wiley and Sons: San Francisco, 2011.
6. CEARLEY, D. et al – Hype Cycle for Application Development – Gartner Group report number G00147982 – Technical Report available at http://www.gartner.com/, 2009.
7. MILLER, M. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. Que Publishing Digital, 2008.
8. SASIKALA, P. Cloud computing: present status and future implications. Available at: <http://www.inderscience.com/storage/f123101246118579.pdf>. Last access: 29 April 2012.
9. ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, v. 1, n. 1, p. 7–18, 2010.
10. NIST. The NIST Definition of Cloud Computing, NIST, 2011.
11. VERAS, M. Cloud computing: new IT architecture. Rio de Janeiro: Brasport, 2012.
12. CEARLEY, D. et al – Hype Cycle for Application Development – Gartner Group report number G00147982 – Technical Report Gartner group. Available at: http://www.gartner.com/, 2009.
13. CHAGANTI, Prabhakar. Cloud computing with Amazon Web Services, Part 1: Introduction, Available at <http://www.ibm.com/developerworks/opensource/library/ar-cloudaws1/index.html>. Last access: 05 April 2013.
14. OPENNEBULA.ORG. Libvirt API 1.4. Available at: <http://opennebula.org/documentation:archives:rel1.4:libvirtapi>. Last access: 24 February 2013.
15. OPENSTACK. What is OpenStack? Available at: <http://docs.openstack.org/cactus/openstack-compute/admin/content/what-is-openstack.html>. Last access: 05 March 2013.