International Journal of Grid Distribution Computing
Vol.7, No.5 (2014), pp.171-180
http://dx.doi.org/10.14257/ijgdc.2014.7.5.16

Research of Enterprise Private Cloud Computing Platform Based on OpenStack

Jiang Yunxia , Zhao Bowen, Wang Shuqi and Sun Dongnan

School of Automation, Harbin Science and Technology University, Harbin, China
hrbustjyx@163.com

Abstract

The demand for computing resources is growing, but the input costs of the enterprise information system construction of the traditional way is too high, and has not the high utilization of the resources. In order to solve this contradiction, the enterprise private cloud platform architecture has been put forward based on open source system. By analyzing the mainstream open source cloud projects, the basis to choose OpenStack has been given. From the perspective of classification of the enterprise personnel, combining with the components of OpenStack, this paper expounds manufacture of the virtual image and distribution, monitor, and use of the virtual machine. At last, the user interface design is also given.

Keywords: *OpenStack, private cloud, virtualization, mirror image*

1. Introduction

For enterprises, creating a set of IT systems needs high investment and maintenance costs. Enterprises want to reduce cost, simplify configuration and network management and integrate idle resources. This is exactly one of powers to promote the vigorous development of the cloud computing. Cloud computing [1] is a mode to on-demand obtain all kinds of IT resources and service from the network. It has changed the mode of providing and management of the traditional IT resources, implemented the intensive share of IT resources, and provided a low cost information solutions for enterprises.

The concept of "cloud" refers to make the IT resource into virtual resource pool, the users can obtain the on-demand IT resources like water, electricity, so as to fully improve the utilization rate of resources. From the view of users, cloud computing platform can be divided into public and private cloud [2]. Public cloud has been deployed in the Internet, run by a third party, and mixes assignment from many different customers within the cloud server storage systems and other infrastructures. Private cloud is owned by a single user, and generally built inside the firewall of enterprises or institutions. It is only provided for internal staff, and is transparent for enterprise external. Compared to the public cloud, private cloud with higher security can satisfy the requirements of enterprises to integrate the existing resources and to decrease the cost of information.

At present, there are many private cloud business solutions on the market, like EC2 of Amazon, Blue Cloud of IBM, Microsoft's Azure, Sun Cloud and so on [3], but commercial solutions are deployed and managed by service providers, with the problems of high price and complex authorization model. However open source cloud computing technology, who uses the open source system to construct the enterprise private cloud platform, has low cost, strong platform scalability. When the platform is designed and come true, it generally keeps the compatibility of mainstream cloud technology, and provides effective conversion interfaces, and avoids the problem of vendor lock-in. Through the contrast of those typical open source

ISSN: 2005-4262 IJGDC

cloud computing projects, a cloud platform architecture based on OpenStack is given, and the method is expounded to build an enterprise private cloud platform by using OpenStack.

2. The Choice of Cloud Computing Projects

2.1. Contrast of Open Source Cloud Computing Projects

There are many open source cloud computing projects, and the mainstreams are OpenNebula, Eucalyptus, OpenStack and CloudStack, *etc*., OpenNebula [4], sponsored by the Institute of European Studies in 2005, is an open implementation of virtualization management layer of the virtual infrastructures and the cloud computing plan. It is divided into three layers, namely, the interface layer, the core layer and the drive layer. The interface layer provides a native XML - RPC interface, at the same time, realizes many kinds of API like EC2, OCCI and OpenNebula Cloud API (OCA) and so on, and provides to users a variety of choices. However, OpenNebula does not adopt SOA design, and not design computing, storage, and network as individual components, so the decoupling degree is not enough.

Eucalyptus [5] is an open source IaaS among the first to try to clone the AWS platform, composed of the cloud controller (CLC), Walrus, cluster controller (CC), storage controller (SC) and the node controller (NC), together collaborating to provide the required cloud services. Components use SOAP messages supporting WS-Security to achieve secure communication. Cluster control let computing service and storage service become independent, but the network is still part of the computing services. Although the network is part independent on code implementation, but overall is not designed as independent services, so the decoupling of overall design is not enough.

CloudStack [6] is an open source IaaS software exploited by Cloud.com, and is divided into community edition and commercial version. In 2011, Citrix System Company acquired Cloud.com, and turned it into a fully open source projects. CloudStack is a typical layered architecture, including the Dashboard/CLI layer, CloudStack API, the core engine layer and computing/storage/network controller layer. But CloudStack itself does not adopt the design of the SOA, and it also does not separate calculate/storage/network parts from the core engine, so it needs to be further strengthened in the aspects of loose coupling and component designing.

OpenStack [7] is an open source cloud computing platform project, which is cooperatively studied by the U.S. national aeronautics and space administration (NASA) and Rackspace. Since 2010, it has released nine versions, and the latest version is the Icehouse. After the Folsom version, storage EBS (Elastic Block Service) and Network are independent from computing Service (Nova), and become the new service (block storage service “Cinder” and net service “Quantum”). Nova calls the cinder and Quantum services through the API. Obviously, OpenStack is the best on SOA and decoupling service components. Therefore, OpenStack is adopted to realize the construction of private cloud in this paper.

2.2 The Fundamental of OpenStack

OpenStack is a project and also open source software, whose purpose is to provide open source software projects to construct, monitor and manage public clouds or private clouds. It has five subprojects, namely Compute, Object Storage, Image Service, Identity and Dashboard, among them, the core subprojects are Compute, Storage, and Image. Compute is the cloud group controller, which provides a tool to deploy cloud, including running instance, managing network, and controlling the access to cloud of users and other projects. Its bottom open source project name is Nova, whose providing software can control the IaaS cloud computing platform, similar to AmazonEC2 and RackspaceCloudServers.

Storage is a scalable object storage system, and Swift component is its concrete implementation. It provides the distributed cloud storage services, supports a variety of applications of its object storage, and realizes the multi-node distributed storage management of cloud files. Image is a mirror storage, query and retrieval system of virtual machines, and Glance component is its concrete implementation. It can be configured in three forms: using OpenStack's object storage mechanism to store the mirror; using the Amazon's simple storage solution (S3) to directly store information; or combine S3 storage and object storage as the connector of S3 access. OpenStack mirror service supports for many virtual machine image formats, including VMware (VMDK), Amazon mirror (AKI, ARI, AMI) and various disk formats supported by VirtualBox. Image metadata container formats include AKI, ARI, AMI, standard OVF format and binary large data [8].

In addition, Identity and Dashboard are implemented by Keystone component and Horizon component, and Keystone provides an unified Identity authentication service, and Horizon provides to the user to use cloud services platform by a Web front-end interface, through which, users can carry out the secret key and roles permissions of mirror instances and cloud storage management, etc. Service relationship among the subprojects is shown in Figure 1. Horizon provides a Web front-end interface for other OpenStack subprojects, which is OpenStack service interface for users and administrators; Keystone provides authentication and other authorized services for OpenStack core services. Nova stores and retrieves the virtual disk (images) and related data in Glance; Glance can store actual virtual disk files in the Swift.

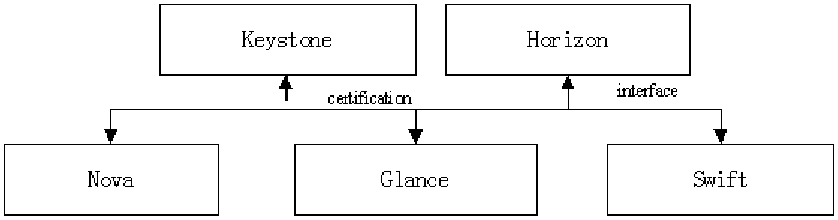


Figure 1. OpenStack Service Relationship

3. Enterprise Private Cloud Platform Architecture

The introduction of cloud computing technology makes the construction investment of the enterprise information system change from the fixed asset into operation, which greatly reduces the initial investment. "Private Cloud" plan eliminates high outsourcing cost of the enterprises, and also increases the efficiency of enterprise assets, and ensures the information security. In order to fully integrate the scattered IT resources, with the service structure of OpenStack, a private cloud computing platform is set up in this paper, as shown in Figure 2.

The concept of OpenStack [9] emphasizes the extensibility and scalability of cloud computing platform as the first element, and any other goals must be compatible with it or optional. This provides to the enterprise with a flexible private cloud building program, as well as provides methods of the access between the private cloud platforms later with public clouds.

Enterprise private cloud platform is built on a computer cluster of multiple physical machines. In order to separate from the upper layers, the underlying physical resources have been virtualized through the Virtual Machine monitoring management tools. Physical resources in the system are computer nodes like enterprise large servers, computing centers and scattered PC, as well as switches and routers and other network physical resources. The main role of virtual machine monitor layer is to virtualized hardware resources of the underlying layer (such as physical servers, *etc.*,), and generates virtual machines with a variety of configuration according to the needs of the upper users, and builds up the logical virtual resource pool. The system can also recycle these virtual resources and realize start, stop, migration and recovery management of the virtual machine. This layer is transparent to users, and there is no difference between the virtual machine and the general physical machine to users. When making virtual images, software and services also can be customized. Namely, in the needs of different people like the technical personnel, accounting personnel, office workers, *etc.*, the corresponding development environments are installed on the virtual machine or related services are started. Users can choose corresponding mirrors and instantiate them, according to their own needs, which avoids the tedious work of installing software one by one, and improves the efficiency.

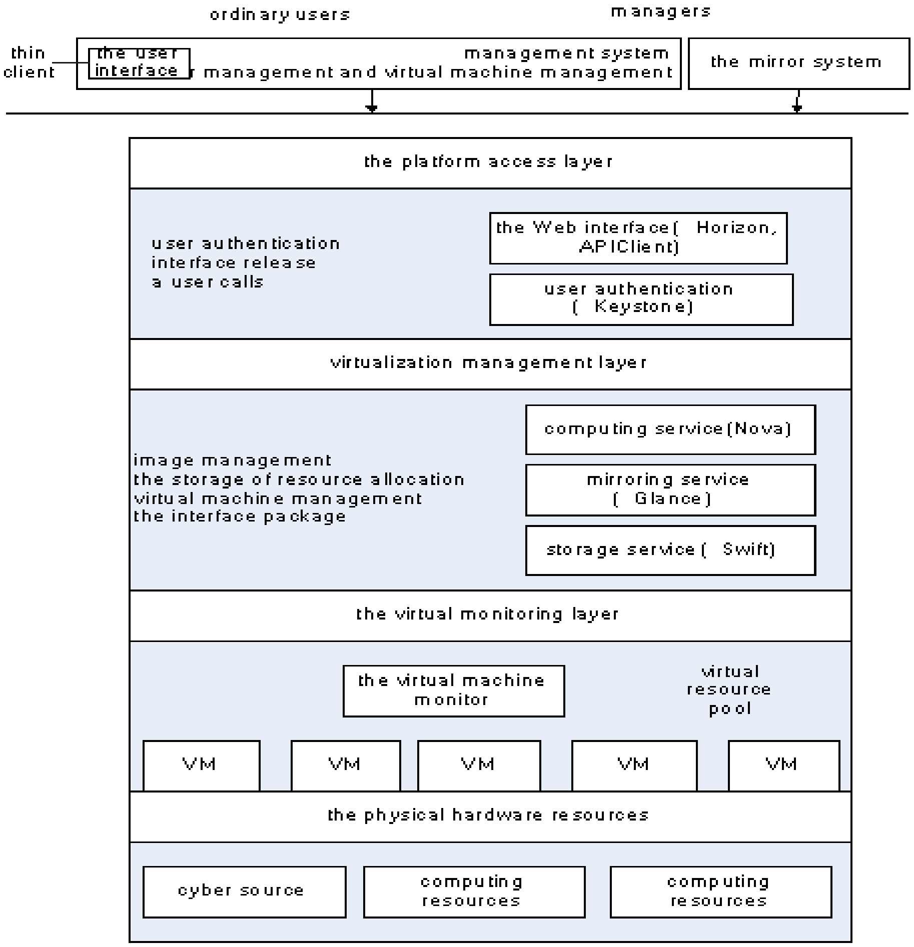

ordinary users
managers
thin client
the user interface
management system
r management and virtual machine management
the mirror system
the platform access layer
user authentication
interface release
a user calls
the Web interface(Horizon, APIClient)
user authentication (Keystone)
virtualization management layer
image management
the storage of resource allocation
virtual machine management
the interface package
computing service(Nova)
mirroring service (Glance)
storage service (Swift)
the virtual monitoring layer
the virtual machine monitor
virtual resource pool
VM
VM
VM
VM
VM
the physical hardware resources
cyber source
computing resources
computing resources

Figure 2. Private Cloud Platform Overall Structure based on OpenStack

Virtualization management layer mainly complete the virtual machine cluster management, virtual machine management, mirror management, and resource allocation, and package their function. The layer is the core part of private cloud management platform, and use OpenStack Compute to complete cloud organization and deployment. The end users interact and compute with OpenStack through the nova-API interface. Nova-compute daemon creates or terminates the virtual machine instance through the hypervisor API; Nova-volume allocates extra persistent storage to virtual machines, and manages creation, connection and separation from persistent volume to the calculation instance, while nova-network realizes the network management. Swift, as the IaaS storage service, docks with OpenStack Compute, and stores image for it. Among them, Proxy Server components accept file upload, data authentication and creating the container; Account Servers manage the account of object storage definition; Container Servers manage the mapping between the container (folder) and the object storage service; Object Servers manage objects in storage nodes. Glance plays a central role in the whole basic cloud setting. It receives API service of requesting mirror from the end users and Nova components, and stores the disk file. Glance-API receives mirror API calls, and Glance-registry completes the storage, processing and retrieval of the data related with mirror.

Platform access layer is the interaction medium between platform and user, which provides a using platform interface for the users and a unified interface for externals and from the perspective of network architecture, the service needs to be accessed to users like other public service API. Horizon (control panel component) is a modular Web application, which provides users and administrators with an interface with the OpenStack service, and realizes the user interact with the system through the Web Service interface. Dashboard (Horizon) must be installed in nodes running Nova API, whose components provides the management interface of the Web form, but due to the control panel components are too simple, system developers need to design the portal system by them, and directly use RESTful API provided by OpenStack to implement the operation and management of each component. Keystone provides configurable directory, policy, tokens and certification service, the main work is to provide authentication services for users of the cloud platform. Users (no matter Dashboard or API Client) need access to Keystone for a list of service and each service address (called Endpoint in OpenStack).

Platform users are divided into users and developers, and the users are divided into administrators and ordinary users, and developers implement applications through API interface provided by OpenStack. Platform architecture designs the management system and mirror system. Management system completes the management of the virtual machine and the user, who sets up perfect management portal for administrators and the mirror system; and the mirror system, realizes virtual mirrors. Ordinary users (hereinafter refers to "user) login the private cloud platform through using a thin client which refers to the operating system of lite version. The thin client only requires a browser and remote desktop function, whose purpose is to speed up the computer start, to avoid unrelated process to consume client resources. It is easy to use, and reduces the costs of the client, and guarantees any network terminal to access to the cloud platform.

4. The Key Issues of Enterprise Cloud Platform

For employees, private cloud platform is a Web application. Its main function is to customize the image template, and distribute and monitor the virtual machine and manage user, *etc*.,.

4.1. Virtual Mirror Production

Administrators need to customize the mirror for private cloud platform, to ensure that the users can get the required virtual instance from the cloud platform. OpenStack itself does not provide the ability of virtualization [10], so it interacts with the virtual machine through the libvirt API, Supporting the virtualization technology such as Xen, VMwherer, KVM and QEMU and so on. The general process of customizing the mirrors: to create a virtual disk→to install and start the selected virtual machine→to make the image file of operating system→to customize applications and services required (optional)→to upload and register the mirror.

In the enterprise cloud, the application software installed by default each is not identical according to the different types of user. For example, product research and development personnel need the professional simulation software; civil servants need office system *etc.*, Therefore, the platform administrator needs to classify the enterprise employees according to their respective functions, defining mirror type templates for each class, as shown in Figure 3, meanwhile, and also needs to open the custom image, allowing the employees to institute the mirrors according to their special requirements.

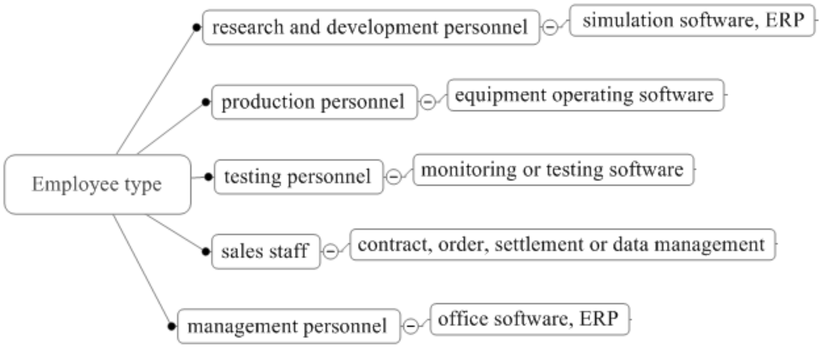

Employee type
research and development personnel
simulation software, ERP
production personnel
equipment operating software
testing personnel
monitoring or testing software
sales staff
contract, order, settlement or data management
management personnel
office software, ERP

Figure 3. The Corresponding Relation of Enterprise Staff and Application

Through the command line or the use of virtualization platform like VirtualBox or KVM, custom image can be built. The first, and vdi image is created, and then it is converted to usable format by using the quemu-img tool in quemu-utils and adding a similar Linux Standard base package and the on-demand application software and services. When a custom virtual machine image is created, we can store the disk image data and image metadata in the mirror service. Here the use of Eucalyptus tools is also a good choice. To install the euca2ools command line tool and ensure the environment contains authentication mechanism required for Eucalyptus to run, and to complete the custom work in the Virtual Box or any other platforms, and then the final files are uploaded and copied in the Linux computer which installs the euca2ools.

4.2. Distribution and Monitoring of Virtual Machines

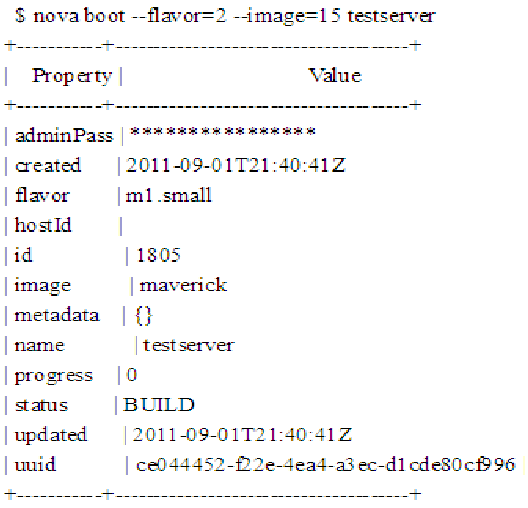
The virtual machine management system provides the management and monitors interface of virtual machine, the administrator can check the detailed information of the distribution of the virtual machine， also can control each virtual machine start-up, shut down, and control of each employee only to operate the virtual machine that create by themselves, realize the monitoring and distribution of virtual resources.

The user can individually apply for the virtual resources, can also be added to the unified distribution queue, receive the uniformly distributed virtual machine by the administrator. As mentioned earlier, the type of virtual machine of the uniform distribution is defined according to the category of employees. Company staffs belong to various departments; each department's function determines the categories of staff. The administrator distributes the virtual machine according to the type, and the department is as the mass unit. The administrator has the ability to count staff categories of each department and the number of virtual machine that each type of staff needs. According to the number of all kinds of virtual machine in the various departments, the virtual machine addresses of each type are divided according to the department. According to the address segment, the administrator distributes virtual machines.

4.3. The Use of Virtual Machine

Thin client used by ordinary users can be a normal PC or any digital terminal that can networking and load the operating system. Thin client browser's default home page is set up as the login page of platform. After the authentication, he can see the list of their specific virtual machine. Different identity authentication has been designed form the view of the users of different types, to provide different virtual resource list. When making a mirror image, VNC or RDP (remote desktop protocol) is installed and open, so that users can use mirror image instance through remote desktop or VNC.

The user can start the instance through the command line. When the instance is started, the user should install the Python-Nova client, which supports the Nova shell command of Compute API interface at the command line. And the user also needs to obtain the user name and password, and set the appropriate environment variables. After that, the user obtains the mirror image information list through the command “$nova image – list”, including name, status, and ID, and selects the required mirror image from it, and finds out the required mirror image and ID related description, and carry on the adjustment. At last, ID value parameters *etc.*, are added into the boot command (creating command); with boot command succeeding, the virtual server has been created. For example, building a small server of 2 GB memory, 20 GB hard drive space, the boot command of the mirror image instance and return information is shown in Figure 4.

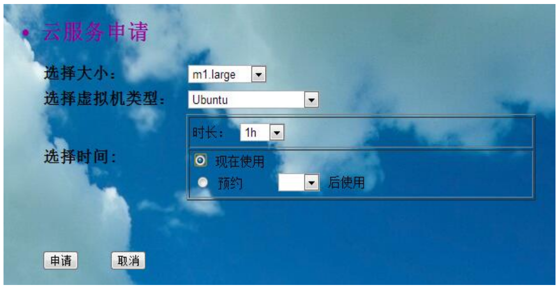


```
 $ nova boot --flavor=2 --image=15 testserver
+----------+------------------------------------+
|   Property |                      Value
+----------+------------------------------------+
| adminPass | ****************
| created    | 2011-09-01T21:40:41Z
| flavor     | m1.small
| hostId     |
| id          | 1805
| image       | maverick
| metadata   | {}
| name        | testserver
| progress   | 0
| status     | BUILD
| updated    | 2011-09-01T21:40:41Z
| uuid       | ce044452-f22e-4ea4-a3ec-d1cde80cf996
+----------+------------------------------------+
```

Figure 4. Returning Information on the Boot Command

5. User Interface Design

The user interacts with the cloud platform through the interface system, to enter the user's all kinds of demand information and control the operation, and automatically call the related subsystem to complete the relevant operation. The employees of enterprise-oriented private cloud almost are ordinary users without IT experience and skills, for this reason, the user interface needs some characteristics, such as easy managed, easy controlled, easy operated and easy monitored and so on. At the same time, the interface must also be friendly and easy to use, and let the user easy to get started and perform various operations. In general, the user applies for the login and the use of virtual machine under the limited condition, through the function of the interface system. According to the design principles and ideas like this, the user interface is divided into 2 modules: the man-machine interface and the input control. The part of the man-machine interface realizes the friendly user-oriented interface, instead of people to enter the demand information. The input control part provides the ability of interacting with each component through RESTfulAPI, like joining the distributed queue, applying virtual machine, and starting up instance *etc.*, Figure 5 and Figure 6 show the interface of applying the virtual machine and the login interface of the virtual machine.



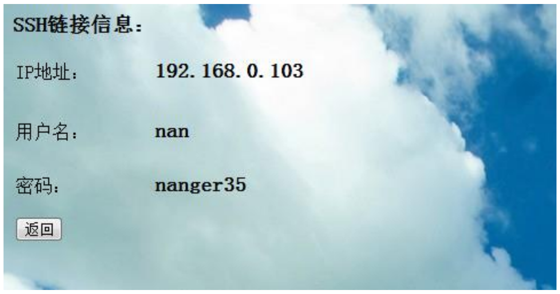
- 云服务申请

选择大小： m1.large

选择虚拟机类型： Ubuntu

选择时间： 时长： 1h
● 现在使用
○ 预约 后使用

Figure 5. Virtual Machine Application Interface



SSH链接信息：

IP地址： 192. 168. 0. 103

用户名： nan

密码： nanger35

Figure 6. Virtual Machine Login Interface

Management system as a unified private cloud portal system to develop, the user interface as its subsystem, the system is on GridsPhere platform, using the Java programming language to realize. Users mainly access management system through the browser, generally need to register, and then to use legitimate user name and password provided by the administrator to log on to the cloud platform. The system provides the various Portlet of applications within the scope of corresponding authority for the user in the browser environment.

6. Result

IT resources are idle and waste in enterprises or institution. In order to improve this phenomenon, by Openstack, the paper designs the enterprise private cloud platform architecture. According to the characteristics of more than class of enterprise personnel, this paper expounds the enterprise resource virtualization mode and the distribution and use of the virtual machine. At last, the user interface system design scheme is given. Hope to be able to reach the elasticity computing and on-demand distributing cloud resources. In addition, the cloud computing solution of an enterprise also needs to consider the diversity of service environment, resource utilization, and other issues. This will be the direction of future efforts.

Acknowledgements

This work is partly supported by the College Students' Innovative Experiment Funds for the Harbin Science and Technology University.

References

[1] “What is OpenStack”, http://liangbo.me/index.php/2012/03/25/openstack-hands-on-lab.
[2] L. Xiaoning, L. Lei, J. Lianwen and L. Desheng, “Constructing a Private Cloud Computing Platform Based on OpenStack”, J. Telecommunications Science, vol. 09, **(2012),** pp. 1-8.
[3] “OpenStack”, **(2012),** http://www.OpenStack.org.cn.
[4] Y. Wencheng, Z. Zhian and L. Wei, “Cloud Computing Resources Management Mechanism Oriented to Virtual Resource”, J. Journal of Northwestern Polytechnical University, vol. 05, **(2010)**, pp. 121-129.
[5] E. Feller, L. Rilling and C. Morin, “Snooze. a scalable and auto-nomic virtual machine management framework for private clouds”, Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Computer Society, Washington, DC, USA, **(2012),** pp. 482–489.
[6] S. Wind, “Open Source Cloud Computing Management Platforms: Introduction, Comparison and Recommendations for Implementation”, IEEE Conference on Open Systems, **(2011)**, pp. 175-179.
[7] M. Mahjoub and A. Mdhaffar, “Comparative Study of the Current Cloud Computing Technologies and Offers”, IEEE First Symposium on Network Cloud Computing and Applications, **(2011)**, pp. 131-134.
[8] J. Yi, W. Weijun, C. Li, L. kai and C. Guiqiang, “Construction of Private Cloud Computing Platform Based on Open Source Software”, J. Telecommunications Science, vol. 01, **(2013),** pp. 68-75.
[9] P. Mell and T. Grance, “The NIST definition of cloud computing (draft)”, NIST Special Publication, vol. 7, no. 800, **(2011),** pp. 145-156.
[10] Q. Zhang, L. Cheng and R. Boutaba, “Cloud Computing State of the Art and Research Challenges”, J. Journal of Internet Services and Applications, vol. 1, no. 1, **(2010),** pp. 7-18.