

Research and Development of “Experiment Cloud” Platform Based on OpenStack

Zhouwuqiang^{1, a}

¹Luoyang Institute of Science and Technology, Henan, Luoyang, China 471023

^a313155832@qq.com

Keywords: Experiment Platform of Cloud Computing; OpenStack; Virtual Desktop; Big Data Application

Abstract. Cloud computing platform suitable for teaching and scientific research is constructed based on OpenStack, to distribute computing resources flexibly and to provide environment support of application development such as virtual desktop and distributed computing. The control nodes of the platform deploy control services for the most part, which operate on two independent virtual machines of CC and NC. By making of the management and deployment of computing nodes using the xCAT tools of control nodes, compute nodes provide hardware and management environment for the operation of the OpenStack virtual machine. The platform can effectively manage cloud resources and provide services of platform level for cloud computing application.

Introduction

With the rapid development of present information technology, cloud computing has become an important hotspot and trend. Cloud computing is based on virtualization technology and forms computing resource pool with a large amount of physical computers and other hardware, so as to provide users with needed computing power, memory capacity and web services, and so on. At present, open source IaaS cloud computing platform has such solutions as OpenStack, AbiCloud, Nimbus, Eucalyputs, CloudStack, and OpenNebula and so on. Among these, OpenStack is undoubtedly the one developing the fastest, which is developed by NASA and Rackspace jointly; as a software with free open source, both enterprises and users can construct their own IaaS (Infrastructure as a Service); it has the features of high expansibility and flexible deployment and can realize the integration and efficient utilization of enterprise resources.

Management technological analysis of OpenStack platform

The effective management of resources and services is a core requirement on the delivery process of cloud computing. According to the differences of function objects, the management can be divided into user management, memory management, network management, computing management, and so on. At present, the management functions concluded in OpenStack platform mainly include: computing management (Nova): providing life cycle management on virtual machines; roll memory management: providing piece memory service and management: authentication (Keystone): providing unified authorization and ID authentication management services; mirror image management (Glance): providing the uploading and memory management of mirror images of virtual machines; network management (Neutron): providing management of the network resources of platform. According to the definition of the five features of cloud computing, a cloud computing platform should contain five characteristics of on-demand services, abroad joining, resource pool management, flexible extension, and measuring service; presently, OpenStack platform has rather mature realization in such three aspects of on-demand services, abroad joining, and resource pool management; however, in such aspects of flexible extension and measuring service, OpenStack still has certain deficiency and space to optimize, which include two aspects of resource monitoring and cluster extension. Resource monitoring is the main chain of the management technology of cloud computing, which provides timely warning disposition for cloud platform and is also the precondition

of such operations of charging and failure recovery. Monitoring objects mainly include the performance data of CPU, memory, memory, network and disk of both physical and virtual machines and the capturing of the start and stop of cloud platform serviced component and abnormal information; based on the captured performance monitoring data, users or managers can be guided to conduct such operations of resource deployment, fault-tolerance processing, and dynamic migration, and so on, so as to realize dynamic load balancing, flexible computing and resource efficient utilization. OpenStack platform doesn't provide complete and mature resource monitoring models; since the version of OpenStackGrizzly, monitoring model Ceilometer is added but this model is not mature, which needs the support of other plugs-in; the model cannot be put into application scenarios yet and it needs the resource monitoring model of its own development. Scalability expresses the capacity of cloud computing on flexible deployment and efficient utilization of resources. Under the circumstance where physical hardware resource of IaaS level is in short, it is necessary to extend OpenStack server cluster; the common means to to utilize physical server to deploy the relevant services of OpenStack and to add it into OpenStack server cluster, so as to realize the extension of hardware resource pool of IaaS level. The deployment of OpenStack nodes involves the installation and allocation debugging of various software, which is rather complicated; artificial deployment has low efficiency and is easy to make mistakes. To deal with the low efficiency and easiness to make mistakes of OpenStack cluster extension under artificial means, the automatic deployment function of OpenStack is taken as the core to add automatic cluster function model for OpenStack, which can enhance the efficiency of flexible extension of cloud platform.

Introduction of Libvirt library

OpenStack supports various kinds of virtualization; in the aspects of KVM and QEMU, Libvirt is adopted as the middle H library to connect and visit virtual machines, to control and manage the state and operation of KVM virtual machine. Libvirt is a set of free C function library of open source supporting the mainstream virtualization tools (such as KVM) of Linux; the API creation tool can be taken to manage operation system or the basic framework of virtual machine Libvirt, which can be seen in Figure 1. Libvirt is a library based on drive program architectural design; the design concept of this framework is that a kind of common API provides services for potential virtual machine monitoring process (such as QEMU, XEN and so on) with common means. Libvirt not only provides the managing function of virtual machines, but also provides the control of memory management, network management and long-distance machines.

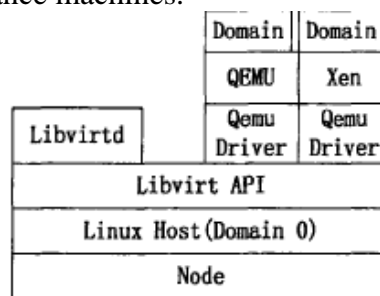


Figure 1 Basic framework figure of Libvirt

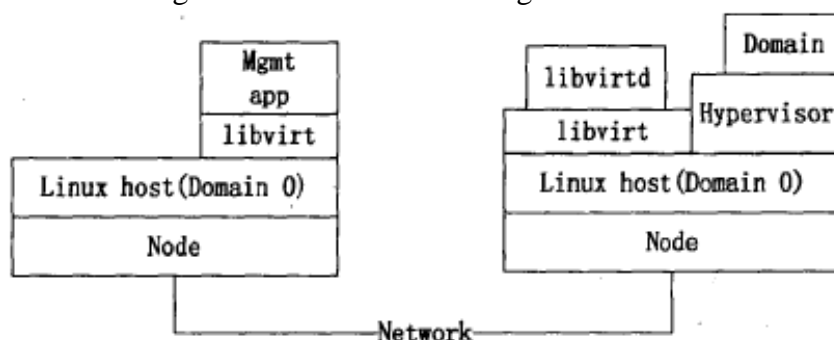


Figure 2 Sketch of Libvirt long-distance control virtual machine

Study and improvement of OpenStack scheduling policy

At the time of building user experimental environment, the background chooses a proper computing node to place user experimental environment through scheduling policy; scheduling policy will influence the possibility of creating experimental environment for users successfully and the frequency of creation failures, which can influence the stability and resource utilization rate of the whole cloud platform. The scheduling process refers to the process that the system needs to choose a target machine to place virtual machine at the time of placing or transferring virtual machines, which has rather high requirements on resource utilization rate. This paper develops certain studies and improvements on OpenStack scheduling policy on the basis of the special requirements.

Improvement on the computing method of weight.

The extremum computing of OpenStack on the mainframe is gained through cost function; the default cost function just takes the free memory of mainframe as measure coefficient; but besides the memory, there are other factors such as CPU and disk influencing the load of mainframe; therefore, this paper puts forward the way to compute the weight of mainframe based on following matrix algorithm.

$$\text{metrics} = \begin{pmatrix} \text{Value1_ram} & \text{Value1_cpu} & \text{Value1_disk} \\ \text{Value2_ram} & \text{Value2_cpu} & \text{Value2_disk} \\ \text{Value3_ram} & \text{Value3_cpu} & \text{Value3_disk} \\ \text{Value4_ram} & \text{Value4_cpu} & \text{Value4_disk} \end{pmatrix}$$

Each mainframe corresponds a line of the matrix; each cost computing number corresponds to a column of the matrix; as the main factors influencing the mainframe load capacity are ram, vcpu and disk, there are 3 cost calculation functions in the matrix: `ram_weigh_object()`, `cpu_weigh_object()`, `disk_weigh_object()`, which corresponds to the H column of matrix, showing the use of ram data, cpu data and disk data as data output. The value of each unit in the matrix is the cost value calculated with the corresponding cost calculation function in the corresponding column; for example, `Value_ram` stands for the cost value of model through cost calculation function `ram_weigh_object()`. The weight calculation formula of each mainframe is as following:

$$\text{weight}_i = \sum(\text{the corresponding Value of the weight factor X of each cost function})$$

The weight factor of each cost function can be adjusted according to the system demands; the choice of weight factor is related to the system demand; for example, cpu occupies the largest weight, and the memory occupies the larger; in the memory system, disk occupies the largest weight, and memory and cpu occupies the larger. This paper mainly emphasizes the placement of virtual machines, and therefore, the choice of weight factors mainly depends on the number of vcpu and memory.

Improvement of scheduling policy.

When none of the host nodes of the cloud platform has the ability to separately place virtual machines, the built-in virtual machine placement strategy of OpenStack will return an error, and the virtual machine request fails; but for the resource pool of cloud platform, there is enough space to put required virtual machine; therefore, the built-in virtual machine placement strategy of OpenStack doesn't utilize the resources to the maximum, which makes the average resource utilization rate `AvjRate` to the maximum; moreover, it is easy to see the circumstance where virtual machine request fails. Therefore, this paper provides the scheduling strategy that is more suitable for the cloud platform of college computers experiments on the basis of the built-in scheduling algorithm of OpenStack; the specific scheduling algorithm is as follows:

Input: required virtual machine `vm`

(1) Gain the available hostlist in cloud plargorm, and rank the average use ratio of calculated mainframe resource from high to low;

(2) Gain the virtual machine list vms on the hostList[0] (the mainframe with the lowest resource utilization rate) and rank from low the high according to the resource amount.

```

(3). while true do
    Foreach vm in vms do
        Rank hostlist from low to high according to hostRate
        If use(vm)<max (hostList[1]) then
            Emigrate vm from hostList[0]
            Imigrate vm to hostList[1]
        else□□
            return□□
        endif□
    if use(vm)□K□max(hostList[0] ) then□□
        Place virtual machine
    Return
    Endif
    End for
    End while

```

The algorithm every time the lowest resource utilization on the host's smallest virtual machine migration to low resource utilization time of host, and then place the virtual machine request to the resource utilization of the lowest on the host. Migration under the strategy of minimum cost, because the lowest resource utilization on a host of the available resources is the most, the fewest W migration of virtual machines can satisfy the current request placement of the virtual machine. In improved virtual machine placement strategy, the cloud platform of resource utilization is always higher than that of before improvement strategy of resource utilization, and can be seen only in the case of resource utilization is higher, the system will start the virtual machine placed failed, before this relative to the improvement of policy system stability is higher.

Test and analysis of OpenStack management platform

Testing environment.

(1) Construct OpenStack platform; the testing environment is based on Vmware10 virtualization software; OpenStack cluster is composed of three Vmware virtual machines which are taken as control node, computing mode and memory node. The disposition of OpenStack nodes can be seen in Table 1.

Table 1 General situation of OpenStack platform

	Control node	Computing node	Memory node
CPU number	2	4	1
Memory /G	2	4	2
Hard disk space /G	20	30	40
Network configuration	Double cards	Double cards	Double cards
Operation system	Ubuntu	Ubuntu	Ubuntu
	12. 04LTS	12. 04LTS	12. 04LTS

(2) Deployment of cloud management platform. The cloud management platform is deployed in the computing nodes of OpenStack; firstly, install the JDK and Tomcat server; then, decompress the software package of cloud management platform into application directory.

Functional test and result interpretation.

Firstly, take test of the cluster extension function of cloud management platform, with the mainframe of IP address 192. 168. 1. 202 as test object; choose the function of adding mainframes automatically in the cloud management platform; after the mainframes are added, use the nova - the manage service list instruction to test the current OpenStack cluster, the function is to list the operation situation of Nova related service elements on all the nodes of OpenStack cluster; it can be seen that the Nova service models on the target mainframes start normally and the target mainframe has been added to the cluster; it can verify the feasibility of the extension cluster.

Then, test the resource monitoring function; log in to the resource monitoring interface, and select the computing nodes in the test environment as test object, to gain the monitoring information data, so as to get the successful access to CPU usage rate, memory usage rate, internet access reception and sending rate, performance changing curve of disk read and write speed. Lastly, choose seven typical cloud computing management functions, and develop the overall function test on cloud management platform and other two OpenStack management platforms of OpenStack— Horizon and Mirantis — Fuel. The test result comparison can be seen in Table 2: in the table “√” indicates that the platform has realized the function successfully, “×” unsuccessfully.

Table 2 Overall function test of cloud management platform

	Cloud management platform	Fuel	Horizon
Management of physical computers	√	√	×
Management of virtual computers	√	×	√
Network management	√	×	√
Cloud memory management	√	×	√
Monitoring of physical computers	√	√	×
Monitoring of virtual computers	√	×	×
Cluster extension	√	√	×

From Table 2, it can be found that cloud management platform has realized all the seven management functions; OpenStack - Horizon lacks in the functions of resource monitoring and cluster extension; Mirantis - Fuel lacks in the virtual resource management functions of virtual resources. Cloud management platform provides OpenStack platform with rather mature management framework, and realizes the optimization of management, which has certain application value and practical significance.

Conclusions

Through the in-depth research and analysis of cloud computing management and development technology, this paper designs the cloud management platform based on OpenStack. On the basis of the basic management operation of OpenStack platform, this paper designs and develops resource monitoring module and cluster extension function module. Through relevant tests, this paper verifies that cloud management platform has certain advancement and application value on the resource management, resource monitoring and cluster extension function. In the future, it is possible to develop further optimization and research on the high availability management and fault tolerance of the cloud platform.

References

- [1] Sun Lei, Shen Subin. A kind of Cloud Management Platform Based on OpenStack [J]. Computer Technology and Development, 2016, 01:185-189.
- [2] Jin Yongxia, Sun Ning. Construction and Application of Cloud Computing Experiment Platform Based on OpenStack [J]. Experimental Techniques and Management, 2016, 06:145-149.
- [3] Yang Zeping, Gu Chunhua, Wan Feng, Luo Fei, Chang Yaohui. Studies on the Innovative Experimental Cloud Platform Based on OpenStack[J].Experimental Techniques and Management, 2016,05:147-150.
- [4] Ma Wenjie. Research and Application of College Experimental Cloud Platform Based on OpenStack [J]. Journal of Suihua University, 2014,12:146-150.
- [5] Sun Juan, Wei Pingping, Li Jiyang, Ran Qibin. Design of Cloud Platform of Computer Experimental Center Under the Framework of OpenStack[J]. Distance Education in China, 2014,09:78-82.