Development of IaaS-based Cloud Co-location and Management System using Open Source Cloud Stack

Chil-Su Kim, HyunKi Ryu, Myung-Jin Jang and Chang-Hyeon Park

Abstract—The weakness of server-based hosting is that it has low usability because it is limited by server unit operation, and it is also difficult to use it completely for all servers or maximize the efficiency of server hosting for all users who cannot reach a mutual consensus. In addition, since many servers are managed by limited manpower, having the entire manpower to work on it is far from being enough, and server hosting-based operation from configuration to problems gives huge burden on manpower. This paper describes that IaaS-based cloud co-location and management system using open source Cloud Stack can be developed to solve such problems, and can help ease the burden of managing manpower by maximizing the usability of server and enhancing the efficiency of hosting management.

Keywords—Cloud, Virtualization, Open Source, Web Services, Ajax, VID.

I. INTRODUCTION

While current IT operation has separate servers for general work or by application service, cloud computing environment is continuously developing the virtualization of server and storage devices that are IT equipment. Users of cloud-based S/W can have hundreds of server and storage devices through the virtualization technology by cloud computing providers, who need to deliver cloud-based S/W as a service concept to users in the virtualized clouding environment [1]. This has led to the acceleration of the development of SaaS (Software as a Service) that are becoming more activated, as the operating system and software platform are expanding from computer-base to web-base [2].

In general, the weakness of server-based hosting is limited by the operation of server unit and has low usability. It is also difficult to use it completely for all servers or even more so to maximize the efficiency of server hosting for all users who cannot reach a mutual consensus. Since a limited number of manpower manages multiple servers, server hosting-based operation from setting to problems lays a huge burden on the manpower [3].

This paper explains that such problems can be solved at once of cloud hosting resources are entirely managed, and describes about an IaaS-based cloud co-location and management system using open source cloud source that can maximize the usability of server and increase the efficiency of hosting management [4]. The system configuration of each part is described in chapter 2 and performance assessment in chapter 3, followed by conclusion in chapter 4.

II. SYSTEM STRUCTURE

A. Main System Configuration

The core of development is that it creates an IaaS system that becomes the base of cloud system and provides a webpage configuration and service that general users can use, as in figure 1. The service can be accessed using a remote desktop function of a PC.

![System configuration for cloud service providers](image)

Fig. 1 System configuration for cloud service providers

B. Development of Cloud Service System

The cloud system configuration is shown on figure 2. It includes host server, which logically combines physical server and network resources, and primary storage server, which will be used as the hard disk (HDD) of virtual machine (VM). Then it consists of ISO file and VHM file, which are the template needed to create a virtual machine, and secondary storage server, which will save the snapshot file that will perform the role of restoration. Also, it consists of a webservice that will operate a cloud homepage for users who will receive public network from the internet through a cloud service, and a PC that runs a management tool that only the cloud system manager can access through a private network only in a domain.
C. Hypervisor of Host Server

The XenServer was selected as the hypervisor of host server. There are various types of hypervisor including VMWare, XenServer, Hyper-v, and KVM, and among them XenServer was used because it was made based on Xen, an open source, can efficiently manage resources such as CPU, HDD, and Memory as much as possible, and is convenient for restoration and duplication. In addition, it supports a wide range of OS (RHEL, CentOS, Oracle Enterprise Linux, SLES 10, etc.) to cover various OS in one client, enables high-performance storage interwork, FC SANs, and iSCIS use, and supports high availability that automatically starts VM during disability.

D. The CloudStack Selection

Apache Software Foundation’s Cloud Stack was selected, as it aims to provide IaaS (Infrastructure as a Service)—a service concept of lending server infra among generally known cloud services—, is widely used among software open sources that were made to enable logically integrate, manage, and control physical server resources, and is the most actively used among utilization products.

E. Cloud Interwork Module

To realize C# interwork module function, the class library project function of C# that can distribute DLL files was developed, and the class library will be referred to easily when making management tool programs in the future.

The functions were inspected by conducting a call and reply test by the types of API managed in the Cloud Stack. With the input parameter of x86-based C# interwork module, the host address (URL or domain name of management server where the Cloud Stack is installed), API key and Secret key of the account that has user and manager authority, API command name to call, and the input values required by the command language were used. This can check success or failure based on the control and API call, and User Interface that can clearly check the replied value was realized. Figure 6 is the main screen of API module test program that can test the status of normal operation of x86-based C# interwork module.
F. Web-based Cross Platform

Generally, web interwork module based on Java Script builds websites in many different ways, and there are differences depending on the development language as well as the web browser. In order to do API call by cloud system in such different kinds of web browsers, a website should be able to be used in any web language and show the same content. This is called cross platform and cross browsing, and any web language can be referred to easily when Java Script is used for such process.

The development of web interwork module function is the list of files to import commonly on the main page using the web interwork module as well as the sub files of /script path, as in figure 7.

![Fig. 7 C# interlocking module test program main screen](image)

G. Development of Management Tool

The login screen needed for user authentication to use the management tool is composed as in figure 8, and only the users with manager authority managed in the cloud system can access it.

![Fig. 8 C# interlocking module test program main screen](image)

The internal structure of management tool was designed to include data section that manages data, process section that processes commands, and I/O section that manages in and output packet. Figure 9 shows the internal structure of management tool represented by schematizing the roles of each section.

![Fig. 9 Management tool & internal logic](image)

The roles of each section are as follows.

The data section is divided into Dictionary Cache, Buffer, and Data Cache in detail. The functions are: security authentication related to management server access, predefined declaratives for mutual API call and reply with the Cloud Stack SW and work details, event, logs for warnings and Dictionary Cache that manages interface for update management, and Data Cache that is practically allocated from Dictionary Cache. Also, the data section buffers the difference between data delivery and processing speed by having Buffer Cache to prevent the bottleneck phenomenon of data used in Data Cache.

The process section is a set of threads that start together when a program runs within the management tool for the actual processing of data and I/O sections. Different processes run for each role, and the types are shown in the process section of management tool internal logic.

I/O section reads and store calculated data and basic information created by a series of work inside the management tool. The data and information are recorded by accessing the DB of management server or as INI file format in a local server.

As for the structure of run-time environment, the data are collected and displayed on the screen by communicating with a physically transferred server as in figure 10, and Cloud Stack management server and web server belong to this. The commands that are mainly processed are response value for API call delivered by the Cloud Stack management server. They have parallel processing because a bottleneck phenomenon occurs in processing commands with higher processing volume as the number of virtual machines in the cloud system increases. In addition, the data delivered to DB socket is used immediately without an additional conversion process. Through such series of work, the virtual machine group can be managed and monitored. Also, the cloud sequence logic is as in figure 11.
C# module interwork was developed by referring to API call library made of X86-BASED c3. Also, NNMAPILib.dll file, which was created through the build process in NNMATILIB class library project, was DLL referred to and added to develop C# module interwork. The namespace was registered as figure 12 to use such additional module.

**H. Viewer Application**

The user authentication interface discerns whether a user trying to use the service is a proper user through an ID and PASSWORD, and server interface selects 1 VM server among the serves the user has and provides an access service on the information and console for the operational status of server.

### III. EXPERIMENTAL RESULT

The performance objective of development technology is as table 1 below.

The performance will be verified by using AIDA64 Extreme Edition and Crystal Disk Mark, which are used the most often for the performance test of CPU and its device in famous hardware venture sites at home and abroad [5][6].

The system performance of CPU will be test by the operation ability of CPU floating point in the cloud system, and a higher value means a better performance. The test environment is as follows.

- Create VM of 10 new accounts on cloud cube homepage and have a test
- Run Windows server 2012 service environment
- Have a test in the basic specifications of 2Core CPU 1GHz, 2G Memory, and 100G Disk
- Measure by using test program AIDA64 Extreme Edition and Crystal Disk Mark

The measurement of CPU performance tests the performance of system with the CPU floating point ability of cloud system, and the higher the value, the more excellent performance it will have.

#### TABLE I

<table>
<thead>
<tr>
<th>Test name</th>
<th>Unit</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Test</td>
<td>FPU VP8</td>
<td>600</td>
</tr>
<tr>
<td>Memory Delay Time</td>
<td>ns</td>
<td>130</td>
</tr>
<tr>
<td>Memory Read Test</td>
<td>Mbps</td>
<td>3,000</td>
</tr>
<tr>
<td>Memory Write Test</td>
<td>Mbps</td>
<td>2,000</td>
</tr>
<tr>
<td>Read Speed(4k)</td>
<td>MB/s</td>
<td>2</td>
</tr>
<tr>
<td>Write Speed(4k)</td>
<td>MB/s</td>
<td>3.5</td>
</tr>
</tbody>
</table>

![Fig. 13 CPU Measurement results](image)

![Fig. 14 Memory delay time measurement results](image)
The results of memory read and write test are as in fig 15. A higher value in the table means a more excellent performance.

![Memory Read/Write Measurement Results](image1)

Then, the measurement results of storage 4K read and write test are shown on fig 15.

![Storage 4K Read/Write Measurement Results](image2)

Fig. 15 Storage 4K read/write measurement results

Then, the measurement results of storage 4K read and write test are shown on fig 15.

IV. CONCLUSION

The system proposed in this paper is a computing that provides virtualized IT resources as a service, and this computing service enables users to borrow and use IT resources (software, storage, server, and network) as much as they need and expand them in real-time depending on service capacity. It is a computing system in which only input/output work are mainly done through a personal device while information analysis and processing, storage, management, and distribution are performed in a third space. The weaknesses of server-based hosting where it was limited by the operation of server unit, had a low usability, and was very difficult to be used completely for all servers and maximize the efficiency of server hosting for all users who could not reach a mutual consensus. In addition, since many servers were managed by limited manpower, having the entire manpower to work on it was far from being enough, and server hosting-based operation from configuration to problem occurrences gave a huge burden on manpower. An IaaS-based cloud co-location and management system, which uses an open source Cloud Stack, could solve such problems at once when managing all hosting resources, could maximize the usability of server, and increase the efficiency of hosting management to provide more relaxation to the managing manpower.

From now on, the cloud system should be adopted quickly with the development of individual system, reduction of equipment purchase cost, etc., to service the centralized software smoothly and strengthen the interwork with the network field and security field.

REFERENCES


Chil-Su Kim received the BS degree in Computer engineering from the Youngdong University and MS, Ph.D Candidate degree in computer engineering from the Yeungnam University in Korea He is the CEO of a software company in Korea. His research interests include multi-media and Artificial Intelligence.