Web Performance Analysis of Open Source Server Virtualization Techniques

Sung-Jae Jung, Yu-Mi Bae and Wooyoung Soh

Dept. of Computer Engineering, Hannam University, Korea
wsoh@hnu.kr

Abstract

Server construction using a server virtualization technique has increased due to its advantages including strengthening security, saving storage and costs, and efficient resource utilization. Particularly when the web server has a severe variation of number of users, server virtualization is used. However, constructing a multi-server with virtualization, compared to a single-server, there can be a drop in performance. In this paper we construct several x86_64 servers based on open source server virtualization tools KVM and VirtualBox, and then analyze their performances using open source analyzing tools ab, httperf, and siege. The performance comparison arises in a single server and multi-server construction based on KVM and VirtualBox with a various number of virtual machines. The test result shows that multi-server construction with a virtualization technique can provide an efficient way of multi-server management in a cloud computing environment.

Keywords: *Sever Virtualization, Web Performance Analysis, Benchmark*

1. Introduction

As web services and smart phones become more popular, web server constructions tend to increase. In the past, the number of web servers was decided according to the access capacity of a physical server and the maximum number of concurrent accessing users. Under the environment of operating multiple servers, there can be a number of operational difficulties including security management, storage space, and operating cost. Furthermore, efficient resource management is increasingly difficult since the number of concurrent accesses is more diverse due to the characteristics of web servers. An efficient solution to these difficulties is to adopt server virtualization technique.

Server virtualization is a technique used to configure and operate a number of virtual servers by dividing the resource of a physical server. It gives a number of advantages including efficient resource management, saving space, strengthening security, and saving costs[1]. Unfortunately, very little study has been done on the performance analysis of server virtualization technique despite its ability to substitute a large amount of physical servers, which is attributed to its efficient resource management.

In this paper we construct several virtual servers into the x86_64 based physical server built on open source server virtualization tools KVM and VirtualBox[2-3], then analyze their performances using open source analyzing tools ab, httperf, and siege[4-6]. We analyze the web performance of the single physical server, multiple servers based on KVM, and multiple servers based on VirtualBox. The web performance of various virtual servers is also evaluated. The results can be used as the basis of constructing multiple servers based on the server virtualization technique.

2. Related Works

2.1. Server Virtualization Techniques

Among the numerous server virtualization tools, XEN, KVM and VirtualBox are the three most popular ones considering their compatibility, performance, preference and usage frequency. These three open source virtualization tools show their own characteristic strength and weakness [7]. XEN requires the installation of both the program and the kernel. It affects other application programs when changing the OS kernel, so we chose only KVM and VirtualBox in this paper.

2.2. Web Performance Analysis Tool

There exist a number of tools for analyzing the performance of both commercially developed and open sourced web servers. In this paper, we selected ab, httperf and siege that are open source tools for measuring web server performance.

2.2.1. ab: ab(Apache HTTP Server Benchmarking tool) is a tool for measuring Apache HTTP server. It is designed to show how the current Apache installation performs, specifically how many requests per second it is capable of serving.

It provides many options such as '–n'(number of requests that can be served per second), '-c'(concurrency number i.e. number of multiple requests performed at a time) and '-t'(time limit – seconds of maximum wait for responses).

One of the most important output of ab is "Time per request" (response time for the request in milliseconds). Others include: Concurrency Level, Time taken for tests, Complete requests, Failed requests, Write errors, Total transferred, HTML transferred, Requests per second, Time per request, Transfer rate.

2.2.2. httperf(http performance measurement tool): Httpperf is a tool for measuring web server performance. A web system under testing consists of a web server, a number of clients, and a network that connects the clients to the server. It provides various options such as --server(the name or IP address of the web site), --rate(specifying the number of HTTP requests/second sent to the web server, indicating the number of concurrent clients accessing the server), and --num-conns(specifying how many total HTTP connections will be made during the test, giving a cumulative number— the higher the number of connections, the longer the test run).

2.2.3. siege: Siege is a tool to measure the web site performance. It can give users the ability to test the resource consumption of the server in a realistic environment. It can support web developers to measure the web server and test how it stands up to load on the network. It supports basic authentication, cookies, HTTP and HTTPS protocols. The user is able to hit a web server with a configurable number of simulated web servers. Siege is very useful when testing Availability and Concurrency of the server. If the Availability is less than 100%, one should pay special attention because it means some of the users could not have access to the server. Concurrency is the result of the processing time of each transaction (number of hits including all authentications) divided by the elapsed time. Concurrency shows the average level of concurrent connections—the higher the Concurrency rate is, the more loads the server has. In other words, the socket is opened to process the new traffic, more time is required for the server to process a transaction, and there is additional traffic for the server to process concurrently. Therefore, the performance of the server goes down.

2.2.4. Items for Performance Analysis: Three items to be used for the performance analysis in this paper are Time per request in ab, Reply time in httperf and Concurrency in siege, respectively. These items are the most relevant in measuring web server performance. The lower the resulting number, the better the server performance.

3. Testing Environment

3.1. Server Constructions

Linux CentOS 6 is installed in a physical server. The process of the performance test is done for a single web server and then the virtualized servers. The virtualized servers are constructed based on KVM and VirtualBox resulting in four virtual machines for the test [8] (Figure 1). Then each server is tested respectively.

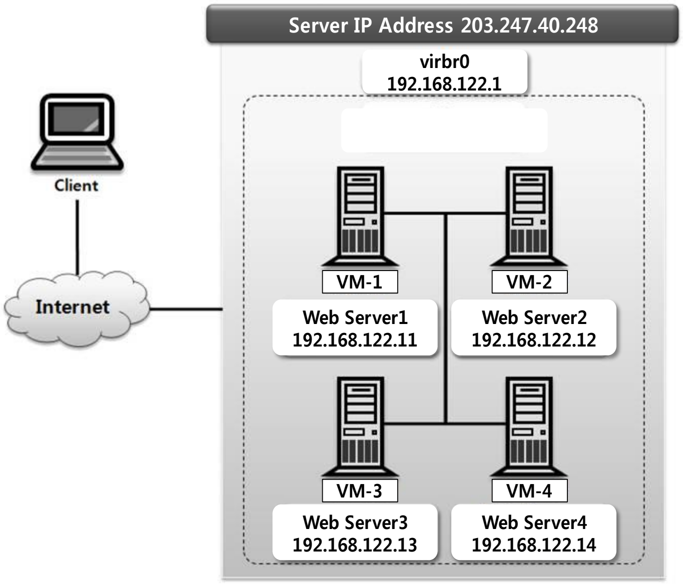

Figure 1. Configuration of Server Virtualization

3.2. Hardware Configuration

In this paper we configured the hardware with x86_64, one of the most popular systems (Table 1).

Table 1. System Hardware Configuration

CPU	Intel Xeon CPU 2.40GHz E5620 4bit Processor
Main Board	Intel S5520HC
MEMORY	8GB(4GB DDR3 PC-10600 ECC/REG * 2)
HDD	Western Digital 750GB Serial-ATA 7200RPM 64M (Model: WD-WCAW30704893)
NIC	Intel 82575EB Gigabit Ethernet

3.3. Configuration of Virtualization

The main system is given an official IP and each of the four virtual machines (VM) is given a fixed private IP address. 100GB of disk space is assigned respectively to store the image file for the virtual machines constructed based on KVM and VirtualBox and the separate partitions, /kvm and /vbox, are created. 1GB RAM and 20 GB disk space are allocated to each of the virtual machines constructed. The disk space of 20 GB allocated to the VM is divided in two areas: 18GB to "/" for Linux area and 2GB for SWAP area.

3.4. Configuration of Software

CentOS 6 is installed in the server as a main OS. In the virtual machines CentOS 6 is also installed and Apache 2.2.15 is installed as a web server program. We used KVM qemu-kvm 0.12.1.2 and VirtualBox 4.1.

4. Test Results

As shown in Table 2 through 4 and Figure 2 through 4, the response time of single server operation is 50 ~ 100 % faster than the operations of the virtual machines by ab, httperf, and siege.

Table 2 shows the performance analysis by ab. The single server operation gives faster time per request than the four virtual machines constructed based on KVM and/or Virtual Box, respectively. Table 2 by ab also shows that the VM constructed by KVM gives faster results than the one by VirtualBox in time per request.

Table 2. "Time per request" (ms) Test by ab

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6	Run #7	Run #8	Run #9	Run #10
Single Server	128.418	128.853	128.828	128.158	127.736	129.133	130.128	128.818	127.536	128.162
KVM	691.821	711.189	717.828	709.838	715.565	714.754	719.436	711.684	713.548	714.144
VirtualBox	799.564	833.694	819.721	826.133	825.125	818.828	832.256	833.496	825.128	828.366

According to the analysis of response time by httperf as shown in Table 3, the single web server construction, VM by KVM and VM by VirtualBox, gives us similar speed rank in reply time.

Table 3. "Reply time" (ms) Test by httper

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6	Run #7	Run #8	Run #9	Run #10
Single Server	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
KVM	1.0	1.0	1.0	1.1	1.0	1.1	1.1	1.0	1.0	1.1
Virtual Box	1.1	1.1	1.1	1.1	1.2	1.2	1.2	1.1	1.2	1.2

From the analysis of the concurrency performance test as shown in Table 4, the single server construction outperforms others, and VM by KVM gives faster response time than VM by VirtualBox in the concurrency test.

Table 4. “Concurrency” Test by siege

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6	Run #7	Run #8	Run #9	Run #10
Single Server	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
KVM	0.12	0.11	0.11	0.12	0.13	0.12	0.12	0.12	0.13	0.12
Virtual Box	0.13	0.13	0.13	0.13	0.14	0.13	0.14	0.14	0.14	0.13

We also tested the response time of the web servers with various numbers of virtual machines constructed by server virtualization technique KVM and VirtualBox. Figure 2 through 4 show the test results with two, three and four virtual machines. As the number of virtual machines increases from two to four, the response time grows slowly not with a continuation but with a convergence appearance. From the analysis of the test we can infer that the response time would not increase above a certain point as the number of virtual machines increases from two to four. The performance deterioration should be insignificant when the maximum number of virtual machines are constructed as the physical sever resources allow.

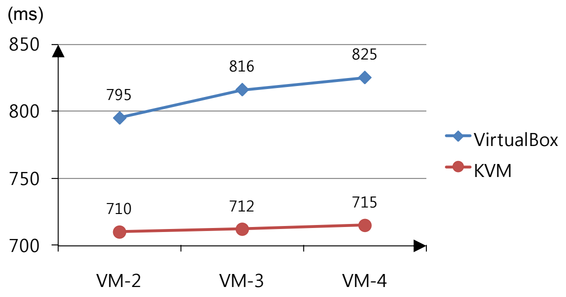

Figure 2. Test Result by ab

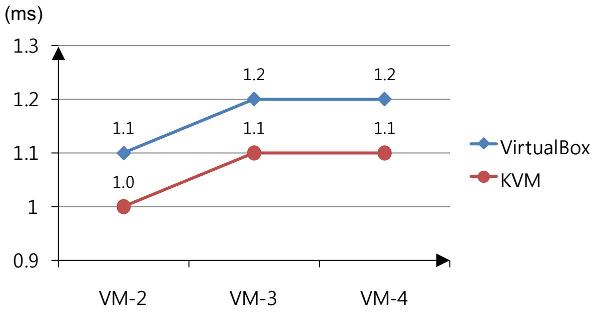

Figure 3. Test Result by httperf

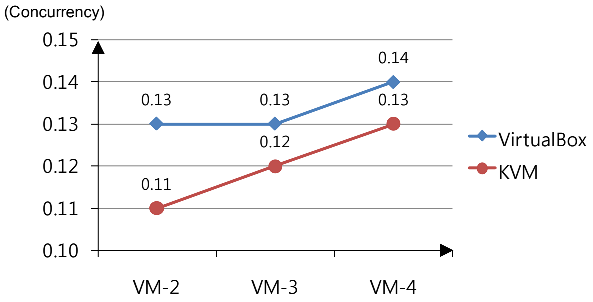

Figure 4. Test Result by siege

5. Conclusions

For an efficient multiple server operation the multi-server construction using server virtualization technique can be an alternative to the single web server construction in a physical system. The former clearly gives slower response time than the latter, but the performance of virtual server depends on the virtualization technique used. As the number of virtual machines increases, the performance of the multi-server configuration deteriorates at the beginning but converges at a certain point.

When using a server virtualization technique in a multiple web server configuration environment, it gives several advantages including strengthening security, efficient resource utilization and saving storage and costs compared with a single server configuration. This is particularly apparent in an environment of severe variation in user numbers. Although one should take into consideration of the performance deterioration, it has a converging point. Therefore, the multi-server construction with a virtualization technique as shown in this paper can provide an efficient way of multi server management in a cloud computing environment.

Acknowledgements

This paper has been supported by 2011 Hannam University Research Fund.

References

[1] Sung-Jae Jung, Yu-Mi Bae, Wooyoung Soh. “A Study on the Secure Enhanced Efficient Web System based on Linux Virtualization”, Journal of Security Engineering, VOL. 7, NO. 4 (2010) 335--350.

[2] Sung-Jae Jung, Yu-Mi Bae, Wooyoung Soh, Kyung Sung. “A Study on the Optimal Server Virtualization for x86 System”, Journal of The Korea Knowledge Information Technology Society, VOL. 5, NO. 5 (2010) 131--139.

[3] KVM, http://www.linux-kvm.org

[4] Oracle VirtualBox, http://www.virtualbox.org

[5] Apache, http://httpd.apache.org

[6] Httperf, http://www.hpl.hp.com/research/linux/httperf

[7] Siege, http://www.joedog.org/index/siege-home

[8] Andrea Chierici and Riccardo Veraldi. A quantitative comparison between xen and kvm. Journal of Physics: Conference Series 219 (2010) 042005.

Authors

Sung-Jae Jung received his Ph.D degree in Computer engineering from Hannam University in 2011. Currently he is working as a senior researcher at an Cloud Computing Company. His research interests include Cloud Computing, Web Clustering System, and Linux Virtualization. Contact him at posein@paran.com.

Yu-Mi Bae completed doctoral research scholarship in Hannam University in 2011. Currently she is preparing a doctoral dissertation. Her research interests include Cloud Computing, Future network and Multimedia. Contact her at yumidw@daum.net.

Wooyoung Soh received his Ph.D degree in Computer Science from Maryland University in 1991. Currently he is a professor in Computer Engineering in Hannam University. His research interests include Computer and Network Security and Cloud Computing. Contact him at wsoh@hnu.kr (Corresponding Author)