Analysis of FTP and Web Server Performance In Open Source Server Virtualization

Reza Mohammadi
Department of Computer Engineering and Information Technology
Shiraz University of Technology, Shiraz, Iran

Seyyed Yahya Nabavi
Department of Computer Engineering and Information Technology
Shiraz University of Technology, Shiraz, Iran

Seyyed Mohsen Emam
Department of Computer Engineering
Islamic Azad University, Khomein, Iran

Abstract—**The use of server virtualization is growing due to its advantages include saving cost and energy, independency, easy control and help maximize server resources. In this paper, we evaluate the effect of virtual server count on server performance. However, by creating a multi-server with virtualization, performance is reduced compared to a single-server. In this paper, we build several servers based on open source server virtualization tools Proxmox, and then analyze their performances using open source analyzing tools httperf, crul-ftp. The performance of single server is compared with multi-server based on Proxmox with a various number of virtual machines. The result shows that multi-server base on virtualization server can improve performance to cost in multi-tenancy environments.**

Keywords-component; Server Virtualization, Server Performance, FTP, Web Server.

I. Introduction

In the computer science literature, virtualization indicates to the act of constructing virtual resources. Virtual resources can be computer hardware, computer software, storage devices, network device, servers, applications, services etc. By using virtualization techniques, application can utilize resources efficiently. Server virtualization refers to create and configure a number of virtual servers on a single physical system. It is a method used to divide the resource of a physical server between numbers of virtual server. Figure 1 shows the server virtualization concept. As seen in figure 1, bare metal hypervisor or hypervisor type 1 is installed on hardware. Hypervisor is a software can creates and runs virtual machine. In server virtualization technique, each virtual machine has its own operating system. Hypervisor also manages the execution of the operating systems of the virtual machines.

Server virtualization technique has benefits such as saving costs, efficient resource utilization, saving place, scalability, manageability and strengthening server security [1][2].

In cloud computing environment server virtualization can reduce the server maintenance expenses by decreasing physical server numbers, server storage spaces, server maintenance labor and server power maintenance [3]. Moreover in data centers, server virtualization techniques can help to power consolidation [4] and increase the flexibility in the provisioning and placement of servers [5].

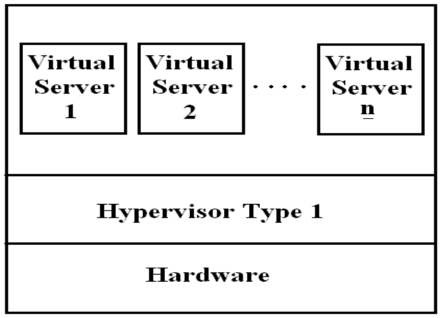

Figure 1. Server virtualization

Web servers and ftp servers are two of useful servers which exist in many organizations, enterprises, institutes and universities to deliver web contents or files to the clients. Large enterprises, institutes or universities can virtualize themselves ftp and web servers by using server virtualization technique to take the advantage of this technique. One of the most important issues in such environment is the performance of servers. An efficient web server is a server that can deliver web pages to clients with minimum response time. Also, an efficient ftp server is a server can deliver computer files to clients uninterruptedly with minimum response time. Server performance measuring tools can help to network administrators to evaluate the performance of servers.

In this paper, several virtual servers are constructed on a single physical server. Proxmox is used as an open source bare-metal hypervisor, and virtual servers are installed on it. Each of virtual servers acts as a web server and FTP server. Operating system of each virtual server is Linux Ubuntu 12.04.4.Performance testing is measured using two open source performance measuring tools (httperf and curl-loader) which are installed on client physical machines.

The rest of this paper is organized as follow. In Section 2, the discussions of the previous related works are presented. Then test environment in detail is presented in Section 3. After that, the performance analysis tools and performance analysis metrics are described in Section 4 and 5. Experimental results are presented in section 6. Finally, conclusions are presented in section 7.

II. RELEVANT STUDIES

Nowadays using virtual system is growing and evaluation of those is base of researches. There are numerous server virtualization tools which some of them are most widespread according to preference, performance and adaptability. There are papers on XEN, KVM and VirtualBox and their strength and weakness [6]. Some of these tools require the installation of both the program and the kernel. It affects other application programs when changing the OS kernel so in [7] only KVM and VirtualBox was chosen.

III. TEST ENVIRONMENT

The architecture of test environment is as figure 2. According to the Figure 1, a type 1 hypervisor is installed on main system, and 6 virtual machines are installed on the hypervisor. Clients can measure the performance of virtual web and FTP servers with httperf and curl-loader tools. In this paper, the main system configuration is in Table 1. Virtual servers are installed and configured on this system.

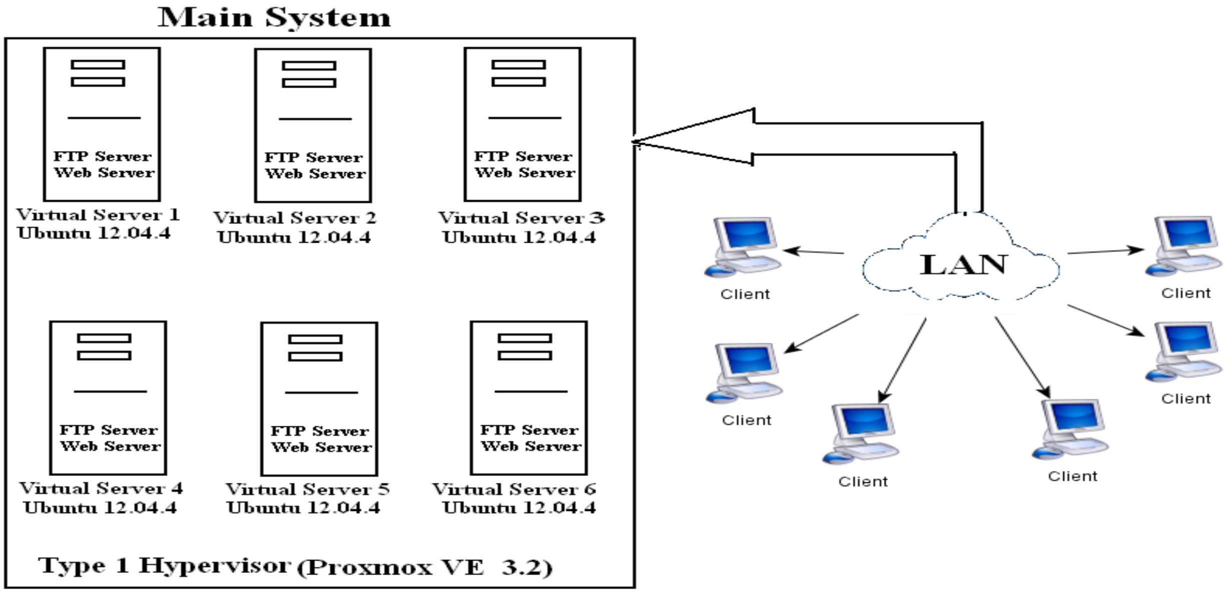

Figure2. The architecture of test environment

Proxmox VE 3.2 [8] is installed as a type 1 hypervisor on main system. Proxmox is an open source bare-metal hypervisor which manages virtualized servers. It is based on KVM (Kernel Virtual Machine) and can support server, storage, and network virtualization. It supports 32/64 bit Windows and Linux virtual servers. Proxmox also supports virtual machine migration and cloning. Proxmox includes web-based management interface which administrators can install virtual server and manage and configure them on it [9]. On each client machine, Linux Ubuntu is installed. For measuring web and FTP servers' performance, httperf and curl-loader tools are installed on Linux Ubuntu.

TABEL I: The main system configuration

CPU	Intel Core i7-3770K @ 3.50GHz, 8MB cache, 4 cores, 8 threads
RAM	4GB
HDD	500GB
NIC	1Gbps

TABEL II: Hardware and software configuration of each virtual machine which installed on Proxmox.

CPU	Intel core2duo
RAM	2GB
HDD	20GB
NIC	1Gbps
OS	Linux Ubuntu 12.04.4
Web Server	Apache Web Server 2.0
FTP Server	Vsftpd

IV. PERFORMANCE ANALYSIS TOOLS

There are many server performance evaluation tools. In this paper, two famous server performance measuring tools are selected. For measuring web server performance httperf tool is selected. httperf is one of useful open source tool for measuring HTTP server performance [10]. It was developed at Hewlett-Packard research laboratories. httperf generates various HTTP workloads for testing web server performance. It can measure metrics such as: connection time, HTTP replay rate, HTTP reply time and etc.

For testing FTP server performance curl-loader tool is selected. Curl-loader is an open source tool for simulating application load and can measure the performance of HTTP/https and FTP servers [11]. It can simulate thousands FTP clients with different source IP addresses and can measure average FTP server delay and average input/output throughput of FTP server.

V. EXPERIMENTAL RESULTS

To demonstrate the effect of using virtual servers based on Proxmox on performance we choice parameters are shown in Table 3.

Tables 4 to 7 illustrate the Single Server operation is better than operations of the virtual machines analyses by httperf and curl-loader.

TABEL III: Performance evaluation parameters

Parameter	Description
Response Time (ms)	The time between sending HTTP request and getting HTTP response in millisecond
Request Rate	The number of HTTP request per second sent to the server
Request Rate	The number of HTTP response per second received from the server
Error Count	The number of HTTP request which is not responded by the server
Throughput (B/s)	The number of received byte per second by the client

TABEL IV: Request Rate, Error Rate and Time per request (ms) Test by httperf

Number of VMs	Single server	2 VM	3 VM	4 VM	5 VM	6 VM
Response Time (ms)	0.4	4	3534	3925	8132	8399
Request Rate (Per Second)	100	100	65	52	27	22
Error Count	0	0	4788	12333	21656	22547

Table 4 shows the performance analysis by httperf. The single server operation gives faster time per request than the one or more virtual machines built based on Proxmox. Table 4 also shows that the request rate in each structure.

We use constant request rate in all test but by the increase in count of VMs, the limitation of hardware resources leads to decrease request rate for each VM and also increase the count of request error. According to the analysis of response time by httperf as shown in Table 5, the single web server construction, in compare of several web server embedded in VMs by Proxmox, gives us better result in reply rate.

TABEL V: Request Rate, Replay Rate Test by httperf

Number of VMs	Single Server	2 VM	3 VM	4 VM	5 VM	6 VM
Request Rate (Per Second)	100	100	65	52	27	12
Reply Rate (Per Second)	100	100	59	44	20	16
Ratio(Req/Res)	100%	100%	91%	85%	74%	72%

We also evaluate the performance of the VMs for file transfer test by curl-loader. In this experience, each VM simulate 80 clients, and each client sends 10 requests for 100kB file. Table 6 illustrates the results. As shown in pervious analyze, the total delay time has a mild increase and looks would not more than a specified amount as the number of virtual machines increases from two to six. Total delay in single server is lower than VMs based on Proxmox, but finally, total delay for VMs converges at a particular level. Table 7 illustrate interesting result, by increasing number of VMs, the variation of throughput was low. These values show that throughput of the system from three to six VMs was improved.

Figure 3 through 6 show the test results with two to six virtual machines. In Figure 3, by increasing the number of virtual machines from two to six, the response time in the single server and two VMs is negligible but for three VMs and more, after big difference it grows slowly. From the analysis of the test we can conclude that the response time would increase the number of virtual machines increases from two to six. Figure 4, illustrates the request and response rate of each test. Because of hardware resource limitation by increasing VMs the request rate decreases but the proportion of request and reply rate is almost constant. In Figure 5, we show that by increasing number of VMs delay time increase but slowly and converges at a certain point so using virtualization don't affect delay time significantly. Figure 6 illustrate the gist of paper, throughput of multi-server in a virtual environment only did not decrease by increasing number of VM but also it was improved a little. We include that unlike the impression virtualization help total throughput of integrated multi-server system.

VI. CONCLUSIONS

The use of virtualization techniques as an alternative method of multiple physical servers has been attracted much attention from researchers. Although the results of this paper indicate that the use of virtual servers based on VM instead of multiple physical systems will have lower efficiency but the increasing number of virtual machines will not a great impact on reducing in performance, and finally their performance converge at a particular level. Hence we can take advantage of virtualization such as saving cost and energy, independency, easy control and help maximize server resources. Obtaining these benefits in multi-server environments is so important. This paper shows that how virtualization can provide an efficient solution to manage multi servers in multi-tenancy environments.

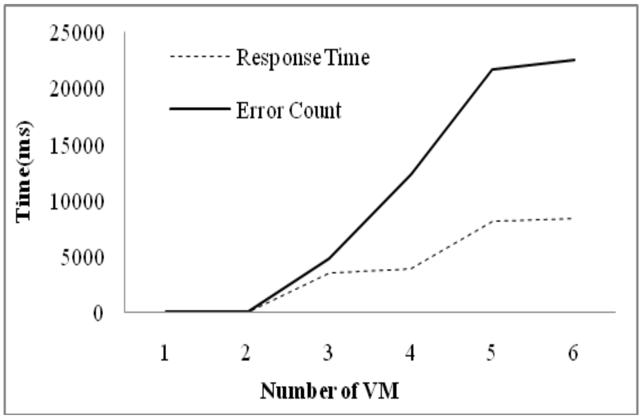

Figure 3: Response Time and Error Count in Single Server and VMs

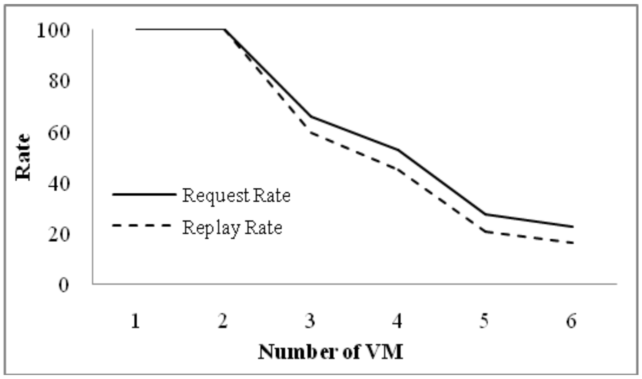

Figure 4: Request Rate and Response Rate in single server and VMs

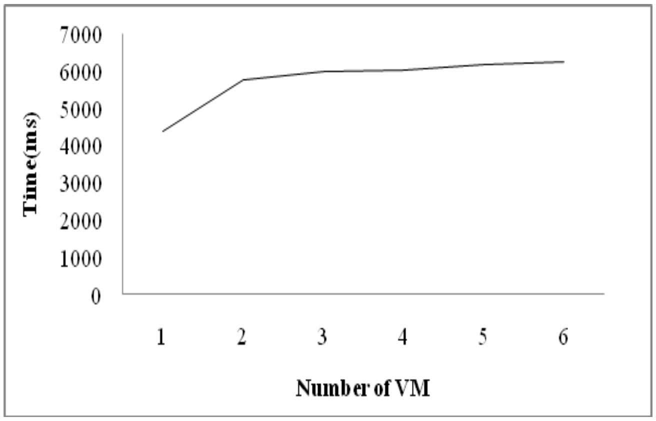

Figure 5: Total delay in Single Server and VMs

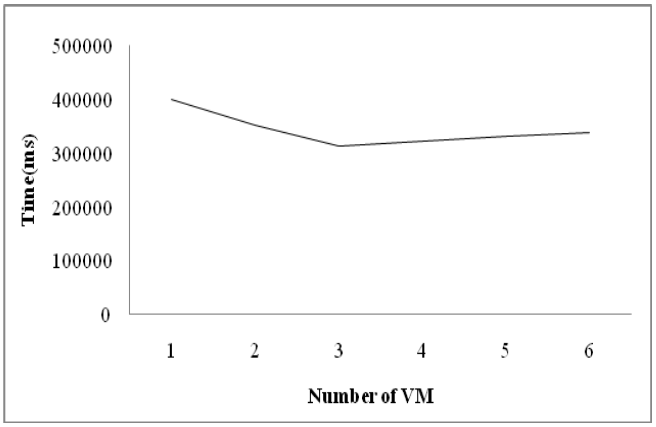

Figure 6: Throughput (in)(B/s) in Single Server and VMs

REFERENCES

[1] Jung, Sung-Jae, Yu-MiBae, and WooyoungSoh."Web Performance Analysis of Open Source Server Virtualization Techniques."International Journal of Multimedia & Ubiquitous Engineering 6.4 (2011).

[2] Daniels, Jeff. "Server virtualization architecture and implementation." Crossroads 16.1 (2009): 8-12.

[3] Baek, Sung-Jin. "Efficient Server Virtualization using Grid Service Infrastructure." JIPS 6.4 (2010): 553-562.

[4] Cardosa, Michael, Madhukar R. Korupolu, and Aameek Singh. "Shares and utilities based power consolidation in virtualized server environments." Integrated Network Management, 2009.IM'09.IFIP/IEEE International Symposium on.IEEE, 2009.

[5] Hyser, Chris. "Autonomic virtual machine placement in the data center." Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189 (2007): 2007-189.

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

[7] Sung-Jae Jung, Yu-Mi Bae, Wooyoung Soh, Kyung Sung. " Web Performance Analysis of Open Source Server Virtualization Techniques" , International Journal of Multimedia and Ubiquitous Engineering. Vol. 6, No. 4, October, 2011.

[8] https://www.proxmox.com/

[9] https://www.proxmox.com/images/download/pve/docs/Proxmox-VE-Datasheet.pdf

[10] http://curl-loader.sourceforge.net/index.html

[11] http://www.hpl.hp.com/research/linux/httperf