Performance Analysis on Network Virtualization Systems

Dr. Kiattichai Atthayuwat[1*]
Santi Homcheun[2]
Benjalak Rattanapahu[3]

[1,2,3]Computer Engineering Department, Faculty of Engineering, Ramkhamhaeng University
E-mail: [1]kiattichai@ru.ac.th

ABSTRACT : The network virtualizations are now starting to be interested by many people and organizations for personal or small and medium enterprise use due to its flexibility, scalability and including low cost implementation. However, there are still some questions whether which one would be more suitable for implementation. This paper presents the analysis and comparison of two most popular network virtualization system, VMware vSphere Hypervisor and Proxmox Virtual Environment in terms of work performance and resource consumption on both standby and full capacity mode. This paper also introduced the design of both physical and virtual network using in most organizations to simulate the use to be as closed as what they would have to do in the real world and to monitor the overall performance on both standby and full capacity mode. Finally, this paper suggests which kind of environment would be more suitable for using these systems.

Keywords : Performance Analysis, Network Virtualization, Proxmox VE, VMware vSphere

1. Introduction

Network Virtualization is the system that allows us to create multiple system and network objects in the way of Virtual Machine (VM) that act like a normal computer or network component. It also allows us to create all the network components, connections, and servers inside one physical system whether it would be a powerful stand-alone server or a cloud system. There are different types of virtualization based on virtualization level, Full Virtualization, Container Virtualization, and Para Virtualization [1].

Full Virtualization [2] is a technique that has the hypervisor installed directly over the underlying hardware. The hypervisor is responsible for loading the guest operating system. Each guest operating system will get all the features of the underlying hardware.

Container Virtualization [2] is a technique in which each operating system kernel is modified to load multiple guest operating systems. The guest operating systems are packed in the form of containers and each container will be allowed to load one by one.

Para Virtualization [2] is a technique in which the guest operating system is aware that they are operating directly on the hypervisor instead of the underlying hardware. The hypervisor will act as the host on which the guest operating systems are loaded. Guest operating system will make the necessary calls to hypervisor for the utilization of hardware resource.

2. Network Virtualization Systems

Two most popular network virtualization systems were selected for this research due to their suitability for personal or small and medium enterprise use, VMware vSphere Hypervisor and Proxmox Virtual Environment. These two applications provide most of the must-have features for those uses with non-paid version.

2.1 VMware vSphere Hypervisor

VMware vSphere [3] leverages the power of virtualization to transform datacenters into simplified cloud computing infrastructures and enables IT organizations to deliver flexible and reliable IT services. VMware vSphere virtualizes and aggregates the underlying physical hardware resources across multiple systems and provides pools of virtual resources to the datacenter.

As a cloud operating system, VMware vSphere manages large collections of infrastructure (such as CPUs, storage, and networking) as a seamless and dynamic operating environment, and also manages the complexity of a datacenter [3].

2.2 Proxmox Virtual Environment

Proxmox Virtual Environment [1] is an open source bare metal environment based on the Debian Linux distribution (also called hypervisor or Virtual Machine Monitor (VMM)) for server virtualization. It allows a user to install different operating systems (for example, Windows, Linux, Unix, and others) on a single computer or a cluster built by grouping computers together. It consists of powerful Kernel-based virtual machines and lightweight OpenVZ containers as an alternative.

3. Performance Evaluation

Performance Measurement and evaluation of network application has been introduced as a following performance metric in [4].

- Server throughput (requests/sec) is the maximum number of successful requests served per second when retrieving web documents.
- Normalized throughput is used as a reference for different configuration setting to make adequate comparison.
- Aggregated throughput (requests/sec) is used as a metric to measure the impact of using varying number of virtual machines on the aggregated throughput performance of a physical host.
- CPU time per execution (micro-seconds/exe) is a performance indicator that shows the average obtained CUP time during each run of the given domain.
- Execution per second measured the number of guest domains being scheduled to run on a physical CPU during one unit time.
- CPU utilization (percent) is measured to understand the CPU resource sharing across Virtual machines running on a single physical machine.
- Network I/O per second (kByte/sec) is the measurement of network I/O traffic transferred to and from a remote web server for the corresponding workload.
- Memory pages exchange per second (pages/sec) is measured to indicate how efficient the I/O processing been.
- Memory pages exchange per execution (pages/sec) shows the average memory pages exchanged during each run of the given domain.

4. Related Works

[2] had researched on the comparison and evaluation of the system performance of three basic hypervisors with one supporting Para virtualization using Xen-PV, other supporting Container virtualization using Open VZ, and the last one supporting Full virtualization using XenServer on various tests conducted by the benchmarking tools.

[4] had researched on the performance measurements and analysis of network I/O applications in virtualized cloud which focused on performance impact of co-locating applications in a virtualized cloud in terms of throughput and resource sharing effectiveness including the impact of idle instances on application running concurrently on the same physical host. Their results showed that performance improvement for cloud consumers can be as high as thirty three percent and the cloud providers can achieve over forty percent performance gain by strategically co-locating network I/O applications.

[5] researched on a performance evaluation framework for virtualized applicances using XenServer virtual machine system and evaluate its performance on four different classes of workload which are VoIP, TCP/IP networking, I/O disk intensive and CPU intensive. Their results showed that the XenSource can isolate hardware resource utilization nicely but significant overhead was found when sharing physical CPU across virtual machines while running multiple virtual appliances concurrently.

[6] had researched on evaluation of two representative virtualization technologies, Xen and OpenVZ, in various configurations. The comparison was done in terms of application performance, resource consumption, scalability, low-level system metrics, and virtualization-specific metrics. Their results showed that the average response time can increase by 4 times in Xen and only one time in OpenVZ. This caused by the higher virtualization overhead in Xen.

[7] researched on performance analysis of network I/O workloads in virtualized data center. They proved that the implementation of virtual machine monitor at that time did not provide sufficient performance isolation to guarantee the effectiveness of resource sharing across multiple virtual machine instances running on a single physical host machine.

[8] had researched on detailed deployment strategies of single-host and multi-host after analyzing Quantum work mechanism and the construction polices of virtual networks. Their results showed

that their virtual network multi-host deployment plan improves virtual network service reliability and efficiency compared with single-node node.

[9] researched on the design of private cloud solution, infrastructure as a service (IaaS), using the OpenStack platform with high availability and a dynamic resource allocation mechanism. They also hosted unified communication as a service in the underlying IaaS and successfully tested voice over IP, video conferencing, voice mail and instant messaging with clients located at the remote site.

Therefore, from all the study above, we decided to come up with the network design that used by regular organization with basic network system including load balance features to two outside ISP connections. The designed metric that will be used to evaluate the performance of the system are filled by measuring the CPU and memory usage percentage, the traffic on both WAN and also the traffic in LAN for standby mode. And for full capacity mode, we decided to measure the actions per minute rate on the each on-service server including, web, mail, file transfer, and database services.

5. Design and Experiments

This research intended to simulate the experimental environment to be closed to the real system. The idea of the design was to have the outside network from Internet Service Provider (ISP), Virtual Network for all the corporate service system, and client network. Two most popular virtualization applications, VMware vSphere Hypervisor and Proxmox Virtual Environment, were placed for tested and compared in term of work performance and resource consumption to point down to where one application could be more suitable than one another.

5.1 Physical Network Design

The physical network was designed to cover all the major systems of most corporates as shown in Figure 1. The local network of the corporate was mainly starting from the virtualization server to the client network. Beyond the virtualization server, two ISP networks were simulated by two physical routers to test the load balancing capabilities. Both routers were connected to a server as a remote server for testing.

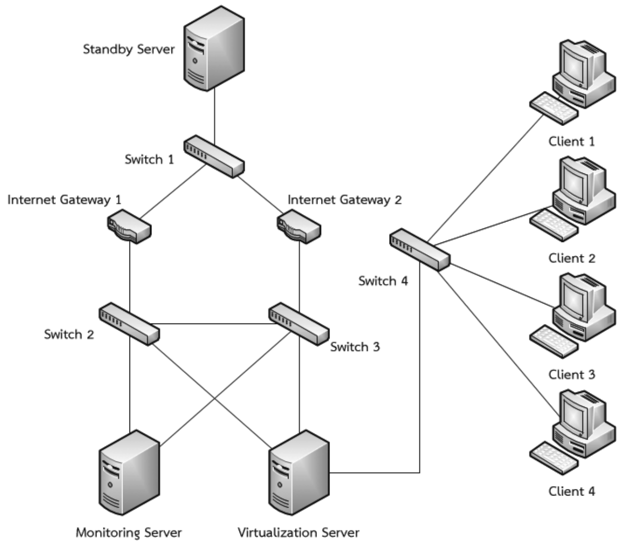

Figure 1 Physical Network Diagram

The specification of each network device and terminal were as shown in Table 1. For the client terminals and standby server, the specification was not specifically defined since they did not involve in processing any packet in the network traffic. Their jobs were only the source and destination of the packet transmission. All of the terminals, except the virtualization server were setup using Ubuntu 14.04 operating system.

Table 1 Physical Network Devices Specification

Terminal	Specification
Virtualization Server	Intel®Core™i7-6700 @ 3.40GHz 32 GB Memory Realtrek PCI GBE Family Controller x3
Monitoring Server	Intel®Core™ 2 Quad Q8400 @ 2.66 GHz 8GB Memory Realtrek PCI GBE Family Controller x2
Standby Server	Intel®Core™ 2 Quad Q8400 @ 2.66 GHz 8GB Memory Realtrek PCI GBE Family Controller x2
Internet Gateway	Cisco 1905 Serial Integrated Services Router
Switch	Cisco Catalyst 2960-24TT-L

5.2 Virtual Network Design

The specification of each device in the virtual network was assigned as shown in Table 2.

Table 2 Virtual Network Devices Specification

Terminal	OS/Applications	Specification
Firewall	Pfsense	2 vCPU 2GB Memory 20GB Harddisk 4 vNIC
VM 1 (Web Server)	Ubuntu 14.04 Apache	2 vCPU 2GB Memory 20GB Harddisk 1 vNIC
VM 2 (Mail Server)	Ubuntu 14.04 Postfix, Dovecot	2 vCPU 2GB Memory 20GB Harddisk 1 vNIC
VM 3 (File Server)	Vsftpd	2 vCPU 2GB Memory 20GB Harddisk 1 vNIC
VM 4 (Database Server)	MySQL	2 vCPU 2GB Memory 20GB Harddisk 1 vNIC

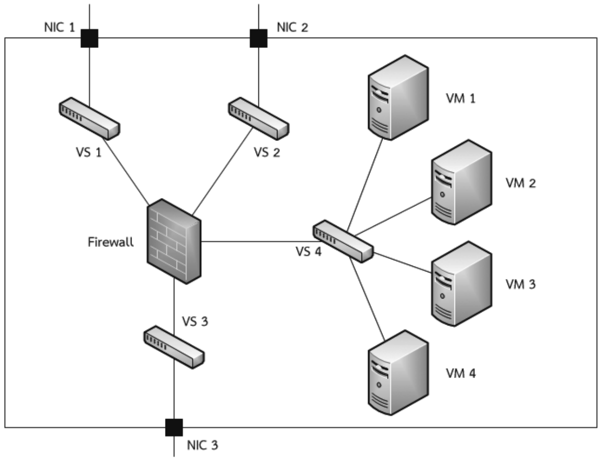

Figure 2 Virtual Network Diagram

The virtual network was designed to cover all the major services of most corporate. The network virtualization application, Proxmox VE and VMware vSphere Hypervisor were all installed in the virtualization server in Figure 1 and set up to have the virtual network as shown in Figure 2. The network interface cards (NIC) shown in the figure were all connected to the physical network as shown in Figure 1 connected via three virtual switches to the Firewall. Four Virtual Machines (VMs) were setup to provide four fundamental corporate network services which were web, mail, file transfer, and database. All of the VMs were connected to the firewall via a virtual switch.

5.3 Testing Scenarios

This research had tested the performance of two network virtualization system, Proxmox VE and VMware vSphere in two categories which are 1) Virtual Network Work Performance and 2) Resource Consumption in two different environments, with and without workload. All the details are as shown in Table 3.

All the scenarios shown in Table 3 would be tested and run to collect data in the same manner. The data collection was done 6 times with 5 minutes paused between each time. During each time the system was run for exactly 20 minutes.

Table 3 Testing Scenarios

Scenarios	Simulation Detail
Work Performance	
- Web Service	Request 2 web different web pages using HTTP with 200 users
- Mail Service	Send and receive mail with 200 users
- File Service	Send and receive file with 47 users
- Database	Search the database with 200 users
	Resource Consumption
- Non Workload	Turn on all the system and do nothing
- Workloaded	Send the packet at 20 Mbps from 4 clients to the remote server

For the virtual network work performance test, Apache JMeter was used to load the command and request to all the servers in the setup Virtual Network. Retrieved result from both Proxmox VE and VMware vSphere were used to determine and analyze the performance comparison.

For the resource consumption test, two applications were used to monitor the resource and traffic consumption. Zabbix was used to monitor the CPU and Memory usage while Cacti were used to monitor the traffic both inbound and outbound for all networks.

6. Result

6.1 Work Performance

The average virtual network work performance comparison value between VMware vSphere and Proxmox VE of each type of service is shown in Figure 3 and Table 4.

Table 4 Virtual Network Work Performance Comparison

Service	VMware	Proxmox
Web (requests/minute)	116.75	145.51
Mail (requests/minute)	26.36	9.47
File (requests/minute)	28.91	27.90
Database (transactions/minute)	99.68	100.63

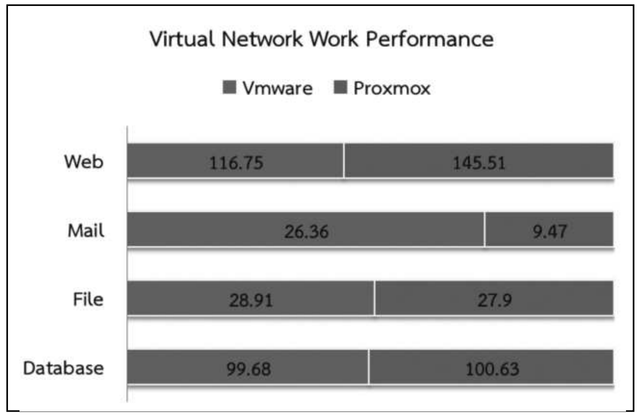

Figure 3 Virtual Network Work Performance Comparisons

6.2 Resource Consumption for Non Workload

Figure 4 and Table 5 show the average collected value of resource consumption in the case of non workload.

Table 5 Resource Consumption for Non Workload

Resources		VMware	Proxmox
CPU usage (percent)		0.88	2.68
Memory usage (percent)		23.72	21.55
Traffic – WAN1	In	770.03	667.12
(Byte/sec)	Out	2,758.69	640.66
Traffic – WAN2	In	206.29	206.33
(Byte/sec)	Out	92.07	92.07
Traffic – LAN	In	2.46	4.20
(Byte/sec)	Out	2.79	5.83

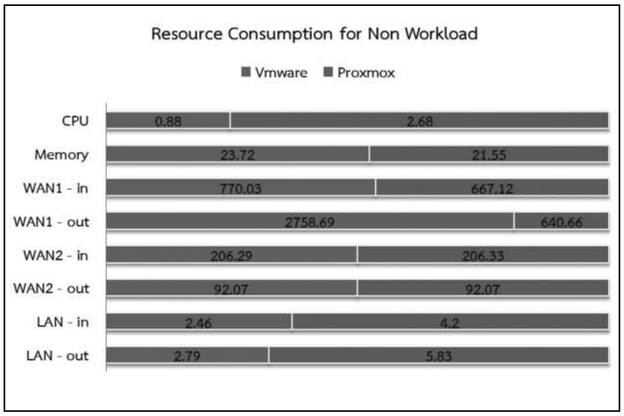

Figure 4 Resource Consumption for Non Workload

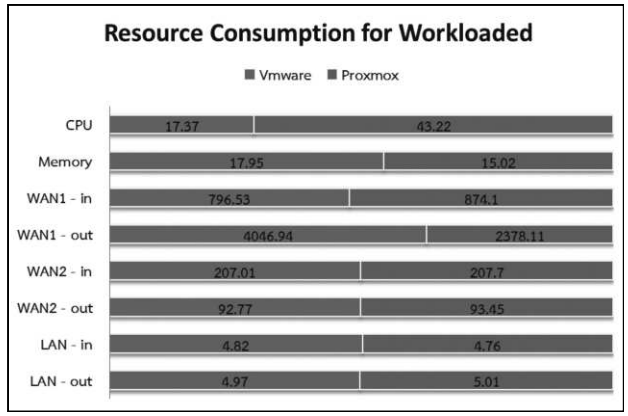

Figure 5 Resource Consumption for Workloaded

6.3 Resource Consumption for Workloaded

Figure 5 and Table 6 show the average collected value of resource consumption in the case of non work-load.

Table 6 Resource Consumption for Workloaded

Resources		VMware	Proxmox
CPU usage (percent)		17.37	43.22
Memory usage (percent)		17.95	15.02
Traffic – WAN1	In	796.53	874.10
(Byte/sec)	Out	4,046.94	2,378.11
Traffic – WAN2	In	207.01	207.70
(Byte/sec)	Out	92.77	93.45
Traffic – LAN	In	4.82	4.76
(Byte/sec)	Out	4.97	5.01

7. Discussion and Conclusion

From all the experimental result, we can see that VMware vSphere Hypervisor is better than Proxmox VE for mail and file service in term of work performance. On the other hand, Proxmox VE is better than VMware vSphere Hypervisor for web and database service. However, for resource consumption on both with and without work load, VMware vSphere Hypervisor is slightly better than Proxmox VE. Therefore, we can conclude that VMware vSphere Hypervisor is better Proxmox VE overall.

Moreover, we would like to point out some of the interesting comparison that we had found during the research. Firstly, we found that Proxmox VE use slightly less storage area than VMware vSphere Hypervisor. Secondly, we also found that, in the version used for this experiment, Proxmox VE has more features for user than what VMware vSphere offered due to the limited license.

Therefore, we would finally advise to use the Proxmox VE for personal or small enterprise due to their ease to use and more helping features. On the other hand, we would recommend using VMware vSphere Hypervisor for a larger system such as small to medium enterprise or even large enterprise where the specification is suitable.

8. Future Works

This experiment only covers some aspect of these network virtualization systems. There are still some other aspects to be covered in the future work. Other network virtualization systems should be selected to be tested as well for complete comparison. In term of application service, more varieties of network service applications should be run and tested in full capacity mode. Testing with the real world environment should be done to confirm the simulation results. The opinion of the experienced network administration should also be included. However, whether which way the experiment should fall into, the tends of the global network environment should be majorly considered.

References

(1) Simon M.C. Cheng, 2014. Proxmox High Availability : Introduce, Design, and Implement High Availability Clusters using Proxmox. Birmingham, UK : Packt Publishing.

(2) Anish B.S., Hareesh M.J., John P.M., Sijo C., and Yedhu S., 2014. System Performance Evaluation of Para Virtualization, Container Virtualization and Full Virtualization using Xen, OpenVZ, and XenServer. 4th International Conference on Advances in Computing and Communications; 247-250.

(3) VMware Inc., 2009. Introduction to VMware vSphere. VMware Inc., EN-000101-0.

(4) Yiduo M., Ling L., Xing P., and Sankaran S., 2010. Performance Measurements and Analysis of Network I/O Applications in Virtualized Cloud. IEEE 3rd International Conference on Cloud Computing; 59-66.

(5) Zhaoqian C., David K., and Kevin M., 2008. Performance Evaluation of Virtual Appliances. First International Workshop on Virtualization Performance: Analysis, Characterization, and Tools (VPACT08).

(6) Pradeep P., Xiaoyun Z., Zhikui W., Sharad S., and Kang G. S., 2007. Performance Evaluation of Virtualization Technologies for Server Consolidation. HP Laboratories Technical Report, HPL-2007-59.

(7) Yiduo et al, 2013. Performance Analysis of Network I/O Workloads in Virtualized Data Centers. IEEE Transactions on Services Computing; Vol. 6, No. 1, January-March 2013; 48-59.

(8) Shaoka Z., Li, Jiahai Y., Cong X., Xiao L., and Shuxiao H., 2013. Deployment and Performance Evaluation of Virtual Network based on OpenStack. International Workshop on Cloud Computing and Information Security (CCIS 2013); 18-22.

(9) Aklilu D.T., Vicky L., and William C., 2015. Design and Implementation of Unified Communications as a Service Based on the OpenStack Cloud Environment. 2015 IEEE International Conference on Computational Intelligence & Communication Technology; 117-122.