Evaluating Performance based on QoS and Resource Management between Virtual Machines for Cloud Platform

Praveen S.M
M. Tech Computer Science Engineering ,Sharda university
Greater Noida (U.P), India

ABSTARCT

With the growth of Information Technology (IT) industry everything is growing rapidly which results in the decrease of efficiency. Virtualization along with cloud computing has become the solution to increase the efficiency. Virtualization technology has earned more and more interest in researcher field because it not only provides to run multiple virtual machines, operating systems on same hardware but also give several benefits such as server consolidation and also severs for security, ease of configuration etc. Here as multiple instances run over the virtual machines the load increase hence there will be great impact over the performances of the whole system. So the Virtual Machines Monitor (VMM) plays an important role as it is the core part of the virtual machine (VM). In this paper, we performed experiments in order to measure the performance of VMM or Hypervisor in different phase for a cloud platform and to understand its results which would be helpful for the developer to make better hypervisor as well as users to make better decision to choose hypervisor based on their interest.

Key words: VMware, VirtualBox, Performance, Benckmark

INTRODUCTION

Along with the booming of virtualization technologies, IT software working groups and vendors (Parallels, VMware, XenSource, and so forth) are developing various performance benchmarking tools to measure the performance of virtualized systems. (Jang et al., 2007)[1]. Back in the days people use to think virtualization is equal to cloud computing but in fact virtualization is considered to be a core important part of cloud computing. Cloud computing in itself is a very large area consists of several other components where virtualization is one of them. It is been said without virtualization cloud computing is nothing.

Virtualization is mainly used for server consolidation hence it is consider as a process of converting one or more physical servers into multiple virtual servers which think and act as physical servers and with the help of management software they can be managed for available resources or the organization's data centres. Virtualization is performed by adding a piece of software to the server that acts as an abstraction layer or kind of an operating system that lies between the physical server and virtual server (virtual machines), this abstraction layer is known as Hypervisor. In virtualization system, resource management and concurrent execution of virtual machines (VMs) are handled by software referred as Virtual Machine Monitor (VMM) or Hypervisor. Once the virtual servers are

created they act as similar to that of physical server, so many applications can be run on the physical server with unaware of each other's presences.

In this paper, we are comparing and evaluating the performance of different virtualization software for Cloud platform. We examine the performance based on the size of the instance and its execution time over multiple virtual machines as well as examining the overall performance while running multiple virtual machines as its performance decreases due to overload.

MATERIALS AND METHODS

ARCHITECTURE

A. *Virtual Machine Monitor (VMM)*

In order to improve system security by providing strong isolation between different guest operating systems (OSs) which can be achieve through virtual machine monitors also referred as hypervisor. VMM or hypervisor is just like an abstract layer or kind of a layer that lays just above the hardware and above that multiple virtual machines running. It basically gives the illusion to each virtual machine that they are the only one physical system running over that hardware. It's a specialized OS that creates multiple virtual processors that behave almost exactly like a real hardware CPU.

Both VMware and VirtualBox lie under the Type II hypervisor i.e. Hosted hypervisor which lies at the second layer and guest operating system runs at the third layer above the hardware. It basically lies at the application layer in this architecture and installed similar to that of other application over the operating system layer that's why it doesn't have direct access to the resources. Here performances of VM depend on the Operating system that is hosting the hypervisor because as here Type II (hosted hypervisor) is used means the hosting operating system manages the access to the resources for the hypervisor. VMware and VirtualBox run on both windows and Linux platform and also support various range of windows and Linux guest. As the no of virtual machines increases its workload also increases which make variation in the amounts of overhead for CPU virtualization.
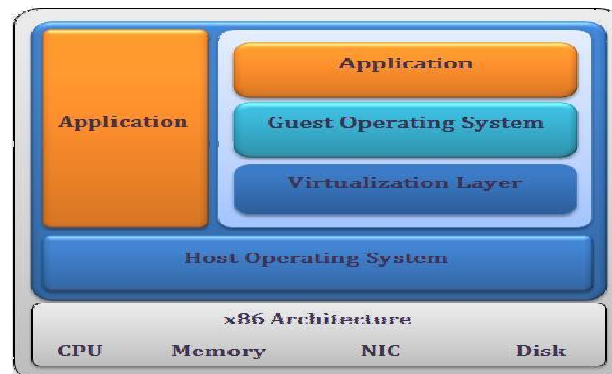

Fig. 1: General Architecture for VMM

Basically CPU virtualization adds less amount of overhead which results in similar performances than that of native host machine. All overhead developed by CPU virtualization through many workloads are not CPU bounded that is rather than executing instruction, spend maximum time waiting for other events such as user interaction, data retrieval etc. because otherwise unused CPU cycles will absorb the virtualization overhead, these workloads typically have throughput similar to native, but potentially with a slight increase in latency.

Here the network address translation (NAT) connections have little greater CPU utilization as its

network performance is comparable to that of bridged-mode network connection. Using virtual machine communication interface sockets (VMCI Sockets) provide good performance results and less CPU overhead on network connection between two virtual machines that lies on the same system.

Linux operating system (Ubuntu) is installed on host machine that runs VMware and VirtualBox. Over which two virtual machines are created. Out of two virtual machine in VirtualBox, one virtual machine containing Linux operating system i.e. Ubuntu 12.04 64-bit OS and another one containing Windows operating system i.e. Windows 8 64-bit OS. Similarly over VMware out of two virtual machines one containing Linux operating system i.e. Ubuntu 12.04 64-bit OS and another virtual machine containing Windows operating system i.e. Windows8 64-bit OS.
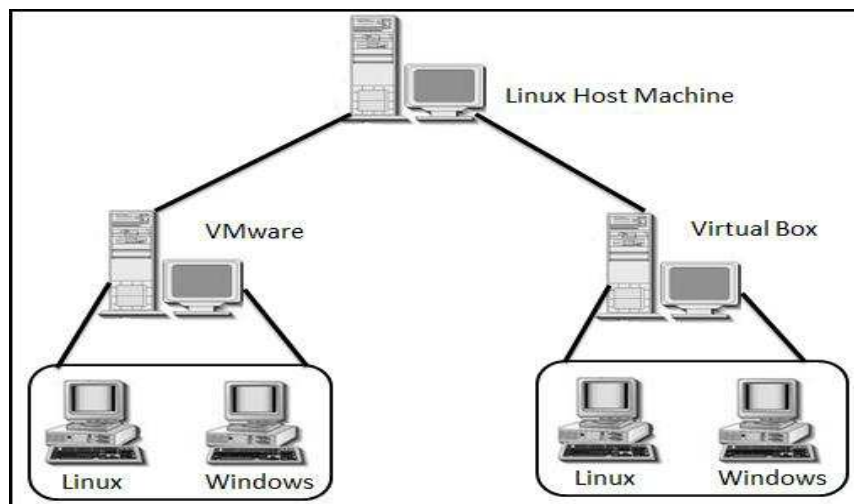

Figure 2: Host machine running VMware and VirtualBox.

BENCHMARK TOOLS

A.   IOZONE

Iozone is disk and filesystem benchmark tool developed by William Norcott which was later enchased by Don Capps that measures variety of file operations. It basically gives the outputs which represent no. of bytes per second that your system can read or write to a file. It tests file I/O performance for various operations such as Read, write, re-read, re-write, read backward, read strided, fread, fwrite, random read, pread, mmap, aio_read, and aio_write. It's a source program written in ANSII C and is compatible in various operating systems. So with this filesystem performance by iozone tool one can choose a platform and operating system that is better balanced.

B.   IPERF

Iperf is an operational measurement benchmark tool used for internet protocol performance measurement which was originally developed by NLANR/DAST. It's basically written in C++ and is available and compatible with both Windows and Linux platform. Here various parameters can be set in order to test the bandwidth and quality of network. With this tool client and server is established and throughput is measured between both the ends which can be bi-directional or unidirectional. It consists of TCP and UDP connection. Here client connects of server, if the connection established is TCP then the output obtained will display the bandwidth of data transfer and if connection is UDP then it displays mainly jitter and loss datagram.

RESULT AND DISCUSSION

PERFORMANCE MEASUREMENT

Here we carried out our experiment using IOZONE and Iperf tool on the virtual machines created over VMware and VirtualBox. The configurations of virtual machines which were created on VMware and VirtualBox are as follow:
Processor: Intel(R) core(TM) i7-2670QM Memory : 2048MB RAM   CPU frequency: 2.20 GHz Number of CPUs: 1 Number of cores: 1 Number of threads: 1

A.   *CPU Performance*

For CPU performance over virtual machines we used the toolkit IOZONE in order to measure the read and write performance over these virtual machines. IOZONE tool is available for both Windows and Linux platform. Installation can be done through executable file available for windows platform and for Linux based platform it's done through terminal by sudo apt-get install iozone3 command. In this first experiment with IOZONE we kept the record size fixed to 1m and made variations in the file size .i.e. increasing file size as 1m, 8m, 32m, 64m, 128m, 512m and 1024m. Through which average throughput per process for initial write and re-write along with read and   re-read   operations can be calculated. Based on the results average throughput per process for initial write, re-writes, read and re-read operations are shown in the graph

From the results initial write and re-write of Ubuntu shows for initial write operation VirtualBox has shown better results than VMware expect for larger file size and for re-write operation VMware had better results than VirtualBox except for the smaller file size. Now same operation carried out with Windows8 platform where both VirtualBox and VMware showing similar results as the file size increases but VirtualBox showed better results than VMware except the larger file size.

So for Ubuntu platform VMware showed better results the VirtualBox but for Windows8 platform VirtualBox had better results than VMware. Similarly for read and re-read operations for Ubuntu except for larger file size VMware showed slight better results than VirtualBox and for Windows8 platform both results of VirtualBox and VMware were better as compared to that obtained in Ubuntu, but here VirtualBox showed much great results as VMware at almost every stage. So for all operations for Ubuntu platform VMware performed better than VirtualBox but for Windows 8 platform VirtualBox had better results than VMware. Overall comparing all operations showed much   better   results   in   Windows   8   platform   compared   to   Ubuntu.
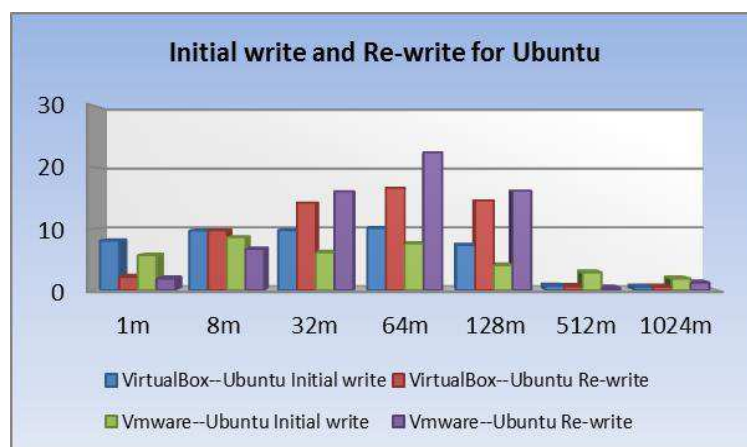

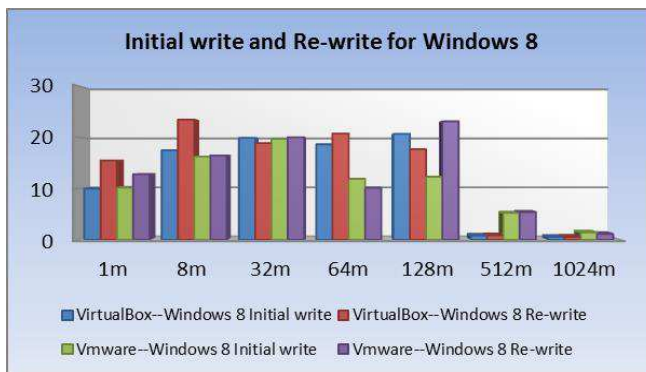Figure 3: Write and Re-write operation for Ubuntu in VirtualBox and VMware

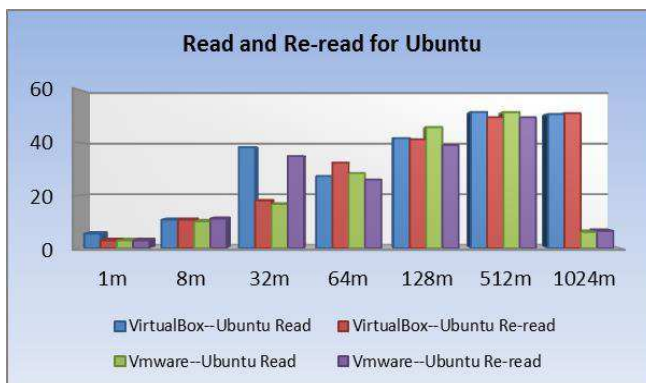Fig. 4: Write and Re-write operation for Windows8 in VirtualBox and VMware

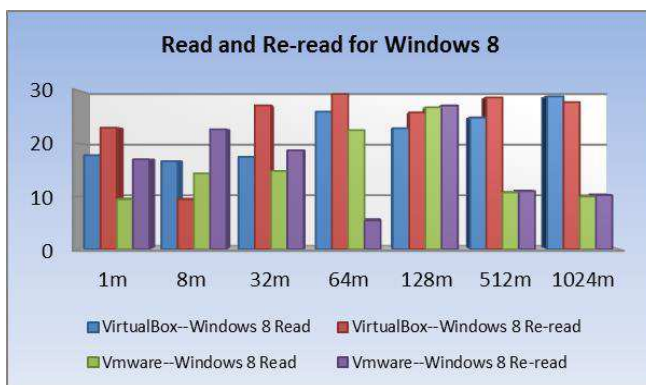Fig. 5 : Read and Re-read operation for Ubuntu in VirtualBox and VMware

Fig. 6: Read and Re-read operation for Windows8 in VirtualBox and VMware

B. Network Performance

Second experiment was carried out for Network performances were Iperf tool was used. Here client and server both were created within the same virtual machines then performances are checked by calculating the bandwidth for transferring of data at certain time intervals. First Iperf tool is executed within Ubuntu inside both VirtualBox and VMware virtual machines and based on the results graphs is drawn which shows that network performance for Ubuntu within VMware is better than that of VirtualBox. Similarly same operation is carried out with the Windows 8 operating system within both VirtualBox and VMware and here for Windows 8 platform VirtualBox had shown better results in network performance than that of VMware.
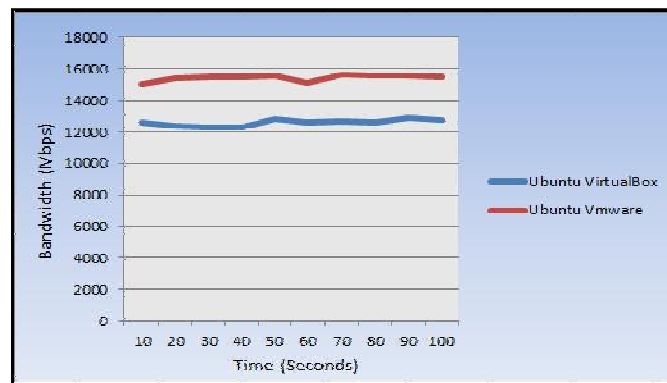
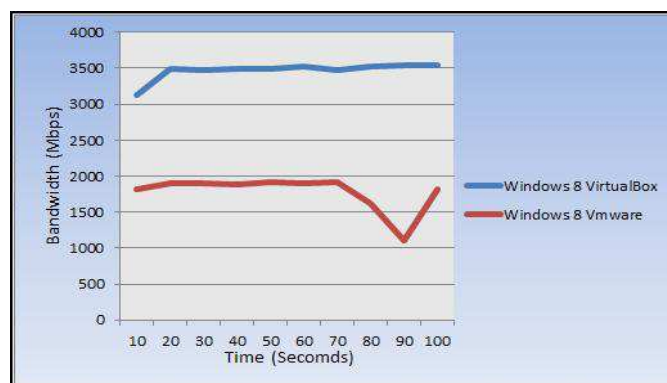Fig. 7: Iperf results-Bandwidth of both Hypervisor in Ubuntu

Fig. 8: Iperf results-Bandwidth of both Hypervisor in Windows8

CONCLUSION

So here we performed our experiments to evaluate the performances over virtual machines with help benchmark tools i.e. IOZONE and Iperf tool. We created virtual machine in both VMware and VirtualBox and carried out operation to evaluate the CPU and Network performances. We measured write and read operation over the files and calculated network response in terms of bandwidth for transfer of data over the operating systems that were installed in virtual machines. Finally comparison was done based on the significant output results that were obtained.

ACKNOWLEDGEMENTS

REFERENCES

[1] Deepak K Damodaran, Biju R Mohan, Vasudevan M S, and Dinesh Naik, "Performance Evaluation of VMware and VirtualBox," *IACSIT Hong Kong Conferences*, vol. IPCSIT vol. 29, **2012**.

[2] Jiyong Jang et al., "A Performance Evaluation Methodology in Virtual Environments," *IEEE Seventh International Conference on Computer and Information Technology*, pp. 351-358, Oct. **2007**.

[3] Vasudevan M.S, Biju R. Mohan, and Deepak K. Damodaran, "Performance Measuring and Comparision of VirtualBox and VMware," *2012 International Conference on Information and Computer Networks (ICICN 2012)*, vol. IPCSIT vol. 27, **2012**.

[4] Jianhua Che, Qinming He, Qinghua Gao, and Dawei Huang, "Performance Measuring and Comparing of Virtual Machine Monitors," *IEEE/IFTP International Conference on Embedded and Ubiquitous Computing*, vol. Volume 2, pp. 381-386, Dec. **2008**.

[5] Xianghua Xu, Feng Zhou & Jian Wan Yucheng Jiang, "Quantifying Performance Properties of Virtual Machine". **2008** IEEE International Symposium on Information Science and Engineering, pp.25-28.

[6] Jiyong Jang et al., "A Performance Evaluation Methodology in Virtual Environments," *IEEE Seventh International Conference on Computer and Information Technology*, pp. 351-358, Oct. **2007**.

[7] Mendel Rosenblum and Tal Garfinkel, "Virtual Machine Monitors: Current Technology and Future Trends," vol. Volume 38, no. Issue 5, pp. 39-47, May **2005**.

[8] Youhui Zhang, Gelin Su, Liang Hong, and Weimin Zheng, "On Virtual-Machine-based Windows File Reads: a Performance Study," *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. Vloume 2, pp. 944-948, Dec. **2008**.

[9] M. Brooks, "A Model of Virtualization," October **2008**, **2008** Dell, Inc.

[10]   R P Goldberg, "Survey of Virtual Machine research," *IEEE Computer Society*, vol. Volume 7, no. 6, pp. 34-45, June **1974**.

[11]   M. Rosenblum and T. Garfinkel, "Virtual machine monitors:Current technology and Future trends," 16 May **2005**.

[12]   Xing Pu et al., "Understanding Performance Interference of I/O Workload in Virtualized Cloud Environments," *IEEE 3rd International Conference on Cloud Computing*, pp. 51-58, July **2010**.