ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

Performance Comparison of Virtualized and Non-Virtualized Guests Machines Based on Their Layer of Abstraction

Parneet Singh
Research Scholar, Department of Computer Engineering
Punjabi University, Patiala
India
parneetsingh62@gmail.com

Gurjit Singh Bhathal
Department of Computer Engineering
Punjabi University, Patiala
India
gurjit.bhathal@gmail.com

Abstract – The main purpose of this thesis is to compare the performance overhead of the virtualization infrastructures KVM and VmWare. All the experiments are carried out by using the Red Hat Enterprise Linux (RHEL) Operating System version 6.1. The study focuses on the performance of disk I/O operations, memory operations and CPU operations. The benchmarking tools used are Iozone for disk I/O, Ram Speed for memory and UnixBench for CPU.

Keywords – RHEL – Red Hat Enterprise Linux, VM – Virtual Machine, Virtualization

I. INTRODUCTION

Virtualization technology is considered the most demanding topic in today's era. Virtualization allows single computer to run multiples operating system simultaneously on a single computer system [1]. Virtualization technology helps the companies to run different services on a single server which enables to reduce the cost of managing more hardware's and usage of resources in more efficient ways. Now a days cloud computing is one of the most hot topic in computer system and virtualization is the key to the cloud computing [2].

There are many reasons to answer the question that why we need to use the virtualization as it has number of financial as well as managerial advantages. There are many challenges that can pop up while deploying the new applications and computer system specially today when modern hardware's are available for commercial and large scale enterprise use. The abstraction from the physical hardware is provided by the virtualization technology [3],that also removes the limitation of running only single operating system on a single hardware.

A. *Basic Concept of Virtualization*

Every virtual machines have a set of following requirements [4]:

- **Equivalence:** The application that is running in virtual machine is just like same as it is running on the hardware without any additional plugin requirement. It must be identical in behavior while running in two different case.
- **Control:** The abstraction layer in between hardware and virtual machines must be controlled and synchronized access of virtual machines to hardware resources.
- **Isolation:** Virtualization technology was developed to ensure the isolation in between virtual machines. The purpose is to ensure stability of crashing one virtual machine should not affect others. Security from compromised virtual machine should not grant access to other virtual machines and data consistency.
- **Performance:** The virtualization overhead that is due to abstraction layer should be minimal, almost close to "Bare Metal" performance.
- **Encapsulation:** Cloning of the virtual machines are easy when they exist in the form of directory of file that also allow easy migration of the virtual machines.

B. *Virtualization Classifications*

- **Full Virtualization**– In full virtualization, virtual machine monitor is also known as virtual machine manager that runs over the host operating system just similar as user application.

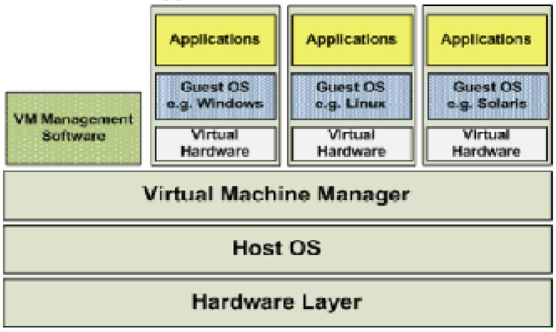

Fig. 1 Full Virtualization

- **OS-Layer or Para Virtualization** –Unlike full virtualization where virtual hardware is created, in operating system layer virtualization, it implements various instances of virtual guest operating system. The virtual machines use the virtualized image of operating system. This kind of virtualization is also known as Single Kernel Image (SKI) or container-based virtualization.

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

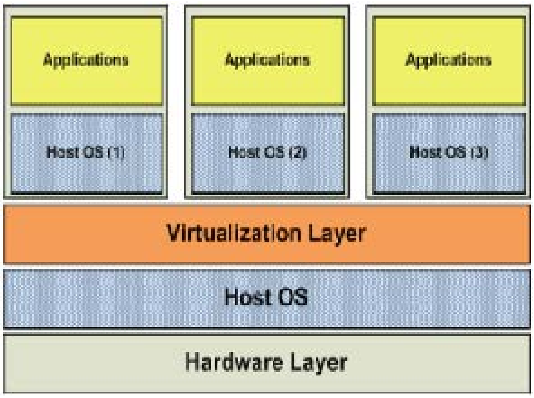

Fig. 2OS Layer Virtualization

- **Hardware-Layer Virtualization --** High performance and isolation is most commonly used by the hardware layer virtualization for server technology. In this category the virtual machine monitor directly runs on hardware that controls the access of the guest operating systems to the hardware resources.

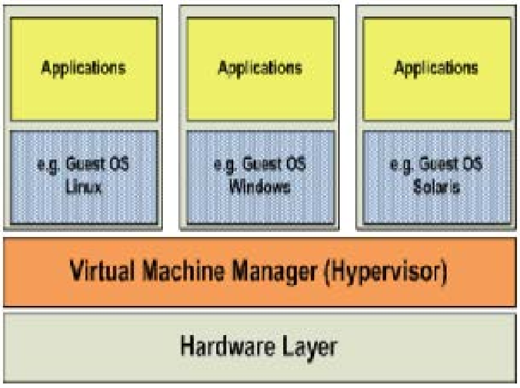

Fig. 3Hardware Layer Virtualization

II. BENCHMARKING

A computerized test for measuring the properties of the particular technology is called benchmarking. The properties might include speed, performance, transfer rate, etc. Benchmarking is important before making decision to select an equipment. The equipment that is going to buy, must be tested before in the same environment and workload as in real working situation. Besides the working situation, it should also has to be tested in worst case situation. It might not always be possible due to non-availability of replicated surrounding environment. This includes the actual data system that is working with. Because of privacy issues of data or huge amount of data replication of the systems data might not be possible. So the artificial workloads are needed for execution and monitoring the benchmark program [5].

While performing benchmarking, the following must be considered:

- **System architecture** 32-bit or 64-bit. x86, Itanium, POWER, etc
- **System size** The number of CPUs in the system under test
- **System configuration** Multiple clustered systems, or a single non-clustered system. Benchmarks for multi-tier applications may use different architectures and operating systems for different tiers.
- **Database size** Range from 100GB to many terabytes
- **Services**Some benchmark includes the cost of 24x7 support, others factors do not support costs into the final result.

A. I/O benchmark tool

- **Flexible I/O --** fio is a tool that will spawn a number of threads or processes doing a particular type of I/O action as specified by the user. The typical use of fio is to write a job file matching the I/O load one wants to simulate.
- **Bonnie++ --** Bonnie++ is a benchmark tool for testing hard disks and file system performances. Bonnie [6] is a micro benchmark, written by Bray in 1988, 1989 and 1996, which measures sequential input and output, creating and reading a file character for character, and in 8 KB chunks and random seeks, in terms of number of bytes per second, on a file.
- **Iozone --** Iozone is a file system benchmark tool. Benchmarks reference various file generation and measurement operations. IOZONE has been ported to multiple machines, and run on multiple operating systems.

B. Memory benchmark tool

- **Bandwidth --** Bandwidth can test the memory in different ways that includes sequential read and write for CPU level 1 and 2 cashes besides the main memory and video memory.
- **RAMSpeed --** RAMSMP is tool written in C and used to benchmark cache and memory to determine the bandwidth of a system's various memory components. RAMSMP can run 18 memory intensive tests at a time and each of them measure a different aspect of the systems memory performance.

C. CPU benchmark tool

- **CPU Free BenchMark --** CPU benchmark is required to know the performance of the CPU specially in virtualized environment when more then one OS routines are competing for CPU.CPU Free BenchMarkgenerate the reliable test results by conducting the 3 identical tests for each CPU part. The data from each identical test is generated in seconds and milliseconds and means of all the three tested is produced. The average of three tests reduce the chance of wrong data collection.
- **UnixBench --** provides performance of Unix systems in various aspects of system performance specifically for CPU. UnixBench produces the results as an index values after comparing the test results with baseline system.UnixBenchis enabled to assess-
  - the performance of the system when running a single task
  - the performance of the system when running multiple tasks
  - the gain from the systems implementation of parallel processing

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

III. APPROACH USED

A. *Method for Conducting the test*

The purpose of this research work is to test the Red Hat Enterprise Linux 6.1 with respect of Disk I/O, memory, and CPU performance in following three cases:

- Bare Metal
- KVM virtual machine
- VmWare virtual machine

The system under test have two SCSI hard disk of 140 GB, one for Bare Metal and KVM and second for VmWare. Being a part of RHEL Kernel, KVM do not need separate hard disk. First of all the RHEL 6.1 is installed on the server with standard installation. In next step install the Iozone 3.3for disk I/O, Ramspeed 2.13.1 for memory and UnixBench 5.1.2for CPU benchmarking.After collecting the data from Bare Metal by using the benchmarking tolls, KVM is activated on RHEL 6.1. After KVM installation, a virtual machine created and RHEL 6.1 OS installed. After installation of benchmarking tool data is collected from KVM virtual machine.

VmWare vSphere Hypervisor (ESXi) 5.0 is installed on second hard disk. To access the VmWare, VShpere client installed and virtual machine is created. RHEL 6.1 along with benchmarking tool installed on VmWare virtual machine for data collection purpose. The the configuration of KVM virtual machine and VmWare Virtual machine was similar. The version of RHEL OS, Iozone, Ram Speed and UnixBench was same on both the virtual machines. To make the test result more reliable and adding authenticity in the data collected, the test was repeated several times. Iozone is repeated 70 time, Ramspeed is repeated 25 times, and UnixBench was repeated three times.

B. *Results*

- **Iozone --** Iozone was used to benchmark the I/O performance of the different guests. Different file sizes of 1 MB, 64 MB, 128 MB, 256 MB, 512 MB and 1 GB were used to calculate the performance with different workload. Following graphs shows the performance of Bare Metal, KVM VM, and VmWare VM.

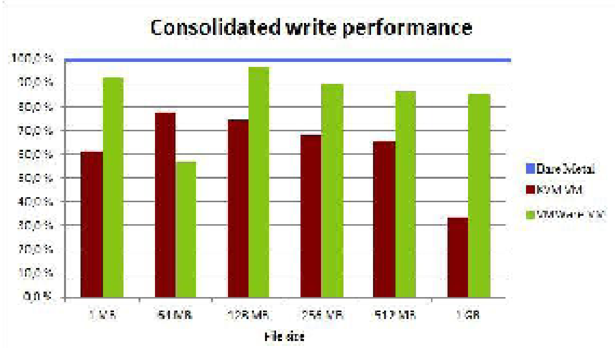

Fig. 4 : Consolidated Write Performance

Fig. 5 : Consolidated Read Performance

The optimum performance of Bare Metal has shown in blue line which is always 100%. VmWare has shown better performance in almost all the file size as compared with KVM. In case of 64 MB file size KVM has shown 78% performance and VmWare was at 57% when compared with Bare Metal 100%. The performance of VmWare in case of 1 GB file size was 85% which was more than twice fast while compared with KVM. In case of 128 MB file size VmWare shows best performance and just behind 2% when comparing with Bare Metal.

- **Ram speed --** For data reliability the ram speed was repeated 25 time with the help of script. Ram speed was tested with exponential of 2 KB block size with maximum of 2 GB. Average from all the test results was calculated and presented in the following graphs-

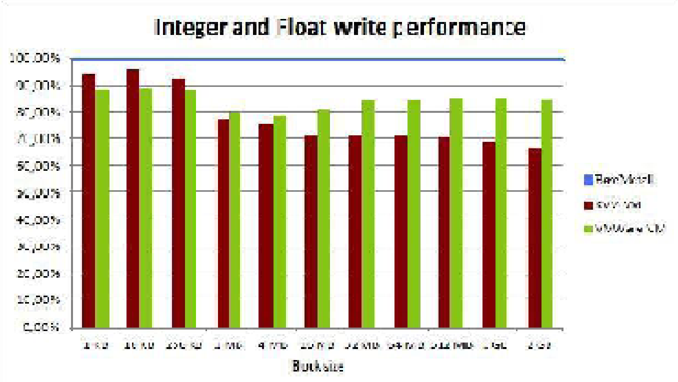

Fig. 6 : Integer and Floating Writing

The writing performance of KVM in smaller block sizes was observed better than VmWare and remained more than 93% of Bare Metal. Whereas the block size larger than 1 MB, the performance of the KVM remained roughly constant with average of 70% of Bare Metal. The performance of VmWare was under 90% of Bare Metal in block size smaller than 1 MB. A continuous rise in performance of VmWare was observed from block size 1 MB and larger.

ISSN 2348 - 2249
Volume-2, Issue-3, July-September, 2014
International Journal for Multi Disciplinary Engineering and Business Management
(IJMDEBM)

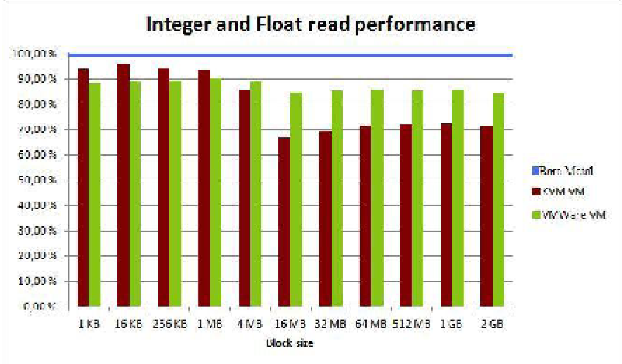

Fig. 7 : Integer and Floating Reading

In case of reading of the ram speed, the consolidated performance 5.21 of KVM was better when compared with VmWare in smaller block size. While in larger block sizes, the VmWare performance was remained 85% of the Bare Metal. KVM in larger block size was remained 70% of the Bare Metal.

- **UnixBench --** UnixBench test system's CPU performance. Performance calculation includes CPU throughput, inter process communication throughput and file system throughput.

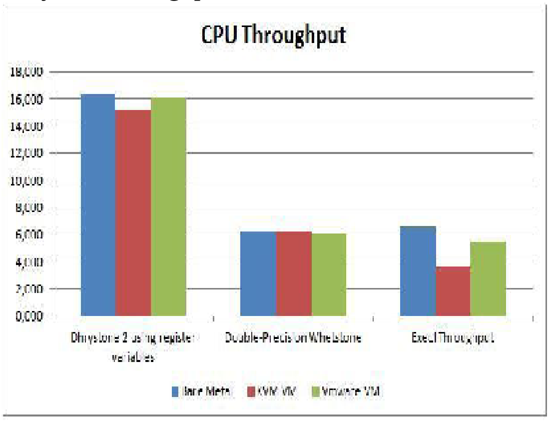

Fig. 8 : UnixBench CPU Throughput

Performance of Bare Metal was remained on the top followed by VmWare that was near to Bare Metal system. Whereas performance of KVM remained little behind than both Bare Metal and VmWare.

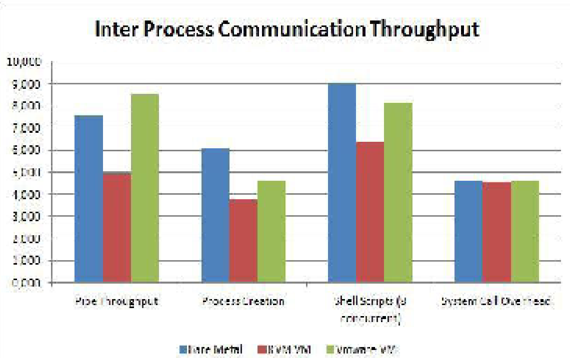

Fig. 9 : UnixBench Inter-process Communication Throughput

In case of system call overhead, the performance of KVM and VmWare was remained identical to the Bare Metal system. In case of pipe throughput, VmWare has shown even better performance than Bare Metal that might be because of some errors. VmWare produced better performance results as compared with KVM.

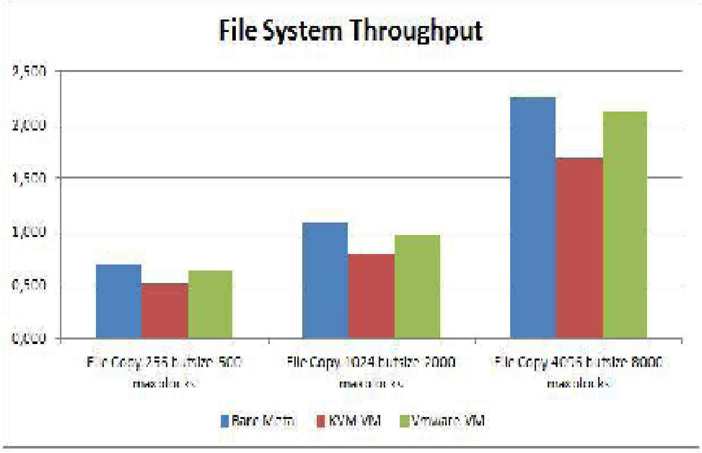

Fig. 10 : UnixBench File System Throughput

VmWare performance better than KVM and very close to Bare Metal. KVM added more overhead in file system throughput and is needed improvement in this area for better results.

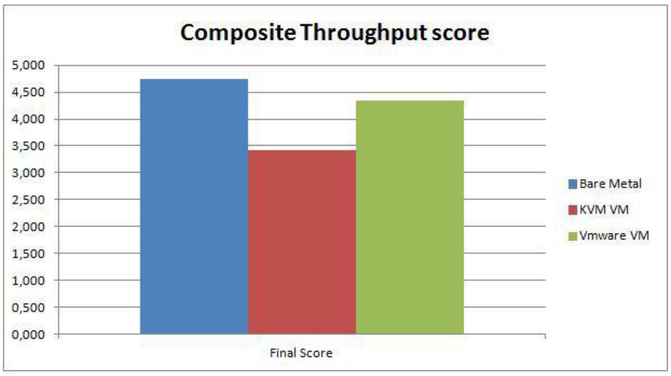

Fig. 11 : UnixBench Composite Throughput Score

In system composite performance VmWare far ahead than KVM and near to Bare Metal guest system.

IV. CONCLUSIONS

While comparing the non-virtualized and virtualized environment, it was observed that the Bare Metal outperform the KVM and VmWare in almost every test. When comparing KVM and VmWare, some interesting results were observed. In case of Iozone, when writing large files, VmWare was more than twice as fast as KVM. In the special case of writing 64 MB files it was observed that KVM was more than 30% better than VmWare. While in reading the VmWare perform better than KVM. In case of smaller file sizes VmWare was 20 to 25% better than KVM. In special case of 1 MB file, VmWare was 40% better than KVM.

While testing the memory performance with ram speed with block size smaller than 4 MB, KVM perform 4 to 7% better than VmWare. In case of block size larger than 4 MB, VmWare perform 15 to 25% better than KVM. In case of other writing VmWare observed 30 to 50% better than KVM.

ISSN 2348 - 2249

Volume-2, Issue-3, July-September, 2014

International Journal for Multi Disciplinary Engineering and Business Management

(IJMDEBM)

CPU performance was measured by using UnixBench. The performance of VmWare was quite close to the performance of Bare Metal. In case of ALU bound processes, no overhead was observed and performance KVM and VmWare was almost similar to Bare Metal. In UnixBench composite throughput score, Bare Metal performs 9% better than VmWare and 29% better than KVM. Whereas VmWare performs 20% better than KVM.

In overall performance, the VmWare perform better than KVM. In some cases VmWare performs twice better than KVM. Whereas, in few cases KVM also gave better results than VmWare.

V. FUTURE SCOPE

Iozone have maximum of 1 GB file size for test. However, it would be interested to benchmark the guests with larger file than 1 GB. Similar with Ram speed, that have maximum of 2 GB block size to test the guest. By using the larger block size, some interesting facts can be unfold.

The performance comparison of the KVM and VmWare was made by using Iozone for I/O, ram speed for memory and UnixBench for cpu. It would also be interested to compare the performance for KVM and VmWare with some other available benchmarking tools. This research was conducted by using the RHEL 6.1. This would also be interesting to compare the performance of the KVM and VmWare by using the different operating systems. The interesting facts can also be unfolded to measure the performance of the KVM and VmWare hypervisors by running several virtual machines. This can help to measures the scalability of the KVM and VmWare.

REFRENCES

[1] Steven J. Vaughan-Nichols, New Approach to Virtualization Is a Lightweight IEEE Computer Society, pp. 12-14, November, 2013.

[2] Herrod, S. A., Systems Research and Development at VmWare ACM SIGOPS Operating Systems Review archive, Vol. 44(4), ACM New York, NY, USA, December 2010.

[3] Crosby, S. and Brown, D., The Virtualization Reality. ACM QUEUE Jr., pp.34-41,December/January 2013-2014.

[4] Popek, G. J. and Goldberg, R. P., Formal requirements for virtualizable third generation architectures. Commun. ACM, Vol. 17, pp. 412-421, July, 1974.

[5] Oppegaard, M. C. A., Evaluation of Performance and Space Utilization When Using Snapshots in the ZFS and Hammer File Systems. Master Thesis, Oslo University College, pp. 29-36, 2009.

[6] Bray, T., Bonnie. http://www.textuality.com/bonnie/, 1996, Accessed 28 Feb,2012

