

Performance Evaluation of VMware and VirtualBox

Deepak K Damodaran¹⁺, Biju R Mohan², Vasudevan M S³ and Dinesh Naik⁴

Information Technology, National Institute of Technology Karnataka, India

Abstract. Virtualization is a framework of dividing the resources of a computer into multiple execution environments. More specific it is a layer of software that provides the illusion of a real machine to multiple instances of virtual machines. Virtualization offers a lot of benefits including flexibility, security, ease to configuration and management, reduction of cost and so forth, but at the same time it also brings a certain degree of performance overhead. Furthermore, Virtual Machine Monitor (VMM) is the core component of virtual machine (VM) system and its effectiveness greatly impacts the performance of whole system. In this paper, we measure and analyze the performance of two virtual machine monitors VMware and VirtualBox using LMBench and IOzone, and provide a quantitative and qualitative comparison of both virtual machine monitors.

Keywords: Virtualization, Virtual machine monitors, Performance.

1. Introduction

Years ago, a problem aroused. How to run multiple operating systems on the same machine at the same time. The solution to this problem was virtual machines. Virtual machine monitor (VMM) the core part of virtual machines sits between one or more operating systems and the hardware and gives the illusion to each running operating system that it controls the machine. Behind the scenes, however the VMM actually is in control of the hardware, and must multiplex running operating systems across the physical resources of the machine. Indeed, the VMM serves as an operating system for the guest operating systems, but at a much lower level; the operating system must still think it is interacting with the physical hardware. Thus, transparency is a major goal of VMM.

Today virtual machine has become popular due to various reasons. Server consolidation [16] is one such reason. It also serves for security, ease of configuration etc. The different types of virtualizations are full virtualization, paravirtualization. Full virtualization is almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified. In paravirtualization a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system. The hardware virtualization support enabled by AMD-V and Intel VT technologies introduces virtualization in the x86 processor architecture itself. The emergence of virtualization hardware assist reduces the need to paravirtualize guest operating systems.

2. Related Work

There exist many works related to the comparison and performance evaluation of different VMM [11, 12, 13] using different benchmarks. Jianhu Che et al.[1] compared VMM like Xen which is an hypervisor based on the x86 platform, KVM (Kernel-based Virtual Machine) using different benchmarks which includes LINPACK, LMBench and IOZONE. Jianhua Che et al.[2] compared Open VZ, KVM and Xen with SPEC CPU2006, LINPACK, Kernel compiling, RAMSPEED, LMBench, IOzone, Bonnie++, NetIO, WebBench, SysBench and SPEC JBB2005 benchmarks and they found that OpenVZ has the best performance and Xen follows OpenVZ with a slight degradation in most experiments, while KVM has apparently lower performance than OpenVZ and Xen.

⁺ Corresponding author.

E-mail address: deepakkakkeel@gmail.com.

Barham et al.[5] introduced the entire concept of Xen and compared it with User-Mode Linux ,VMware, and Base Linux using SPECcpu2000, OSDB, dbench and SPECweb99. Clark et al.[10] repeated the performance analysis of Xen done in [5] but extended the analysis by including comparing XenLinux on x86 to an IBM zServer. Deshane et al. [6] performed quantitative comparison of Xen and KVM focusing on overall performance, performance isolation, and scalability with kernel compile test, I/O intensive test. Xianghua Xu et al. [2] performed a quantitative comparison of Xen, KVM and VMware focusing on the overall performance, performance isolation, and scalability of virtual machines using Netperf, IOZONE, and SysBench. Most of the researches were focusing on the VMM like Xen and KVM; here we introduce VMware and Virtual Box for performance comparison which has not been considered in any of the experiments.

3. Virtual Machine Monitor

VMM serves as an operating system for operating systems, but at a much lower level. VMM sits between one or more operating systems and the hardware and gives the illusion to each running OS that it controls the machine. Here we describe the virtual machine monitors VMware and VirtualBox.

3.1. VMware

A hosted x86 virtualization monitor [15, 17] which can run a guest operating system unmodified with some performance loss. The x86 architecture offers four levels of privilege known as Ring 0, 1, 2 and 3 to operating systems and applications to manage access to the computer hardware. While user level applications typically run in Ring 3, the operating system needs to have direct access to the memory and hardware and must execute its privileged instructions in Ring 0. Virtualizing the x86 architecture requires placing a virtualization layer under the operating system to create and manage the virtual machines that deliver shared resources. Some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0. The difficulty in trapping and translating these sensitive and privileged instruction requests at runtime was the challenge that originally made x86 architecture virtualization look impossible. VMware resolved the challenge by developing binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance, while moving the operating system to a user level ring with greater privilege than applications in Ring 3 but less privilege than the virtual machine monitor in Ring 0.

3.2. VirtualBox

Oracle VM VirtualBox is an x86 cross platform open source virtualization software package [18], now developed by Oracle Corporation as part of its family of virtualization products.

VirtualBox is a so-called "hosted" hypervisor. To a very large degree, VirtualBox is functionally identical on all of the host platforms, and the same file and image formats are used. The guest OS code, running in ring 0, is reconfigured to execute in ring 1 on the host hardware. Because this code contains many privileged instructions which cannot run natively in ring 1, VirtualBox employs a Code Scanning and Analysis Manager (CSAM) to scan the ring 0 code recursively before its first execution to identify problematic instructions and then calls the Patch Manager (PATM) to perform in-situ patching. This replaces the instruction with a jump to a VM-safe equivalent compiled code fragment in hypervisor memory. The guest user-mode code, running in the ring 3, is generally run directly on the host hardware at ring 3. In hardware assisted emulation VirtualBox supports both Intel's VT-x and AMD's AMD-V hardware virtualization. Making use of these facilities, VirtualBox can run each guest VM in its own separate address space; the guest OS ring 0 codes runs on the host at ring 0 in VMX non-root mode rather than in ring 1. A newer feature called nested paging implements some memory management in hardware, which can greatly accelerate hardware virtualization since these tasks no longer need to be performed by the virtualization software.

4. Testing benchmarks

This section describes about two benchmarks that are used in the following experiments.

4.1. LMBench

LMBench [14, 19] is an open micro-benchmark suite to test physical hardware and basic processing units of operating systems and common. LMBench mainly focuses on the bandwidth and latency of data movement between processor, cache, memory, disk and network. Specifically, LMBench measures not only the bandwidth of cached file read, memory read, memory write, memory copy (bcopy and hand-unrolled loop), pipe and TCP, but also the latency of context switching, networking(connection establishment, pipe, TCP, UDP and RPC hot potato), file system creating and deleting, process creation(fork, fork+exec and system), signal handling, system call, etc. Besides the above, LMBench may also test the processor clock rate calculation. LMBench does not take any advantage of multiprocessor features to get the performance data.

4.2. IOZONE

IOzone [20] is a free file system benchmark. IOzone takes various file operations as basic workloads to test the I/O performance of file systems and allows diverse testing modes from automated to partial. The file operations include read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read/write, pread/pwrite variants, aio read, aio write and mmap. IOzone owns numerous characteristics. Its 64-bit's compatible source program is written in ANSII C and able to measure the performance of POSIX async I/O, normal and mmap file I/O, single and multiple stream, POSIX pthreads, multi-process and size-configurable processor cache, I/O latency data for plots, stonewalling in throughput tests to eliminate straggler effects.

5. Experiments and Performance Study

Here we have done the comparison of VMware and VirtualBox by benchmarking virtual machines using LMBench and IOZONE.

The virtual machines were created on VMware and VirtualBox, the configurations were

Processor: Intel i3.

Memory: 1GB

CPU frequency: 1.344 GHz

Number of CPUs: 1

Number of cores: 1

Number of threads: 1

4 virtual machines were created .Two on VMware and two on VirtualBox each with the above configuration. Out of the two virtual machines on VMware one contain Windows XP Dark edition as guest OS and the other one contains Ubuntu 10.10 as guest OS. Out of the two virtual machines on VirtualBox one contain Windows XP Dark edition as guest OS and the other one contains Ubuntu 10.10 as guest OS. Each virtual machine was created with Ubuntu 10.10 as the host operating system.

For probing into memory virtualization we have used LMBench to benchmark virtualized Ubuntu in VMware and Ubuntu in VirtualBox. Here we have focused on the latency and bandwidth of system performance. First we have measured the memory read and write bandwidths of the two virtual machines. Memory read measures the time to read data into processor and memory write measures the time to write data into memory. The results are shown in Fig.1.In all the following figures the size specification “k” indicate kilobytes and “m” indicates megabytes. Fig.1 shows that the memory read for VMware is higher than VirtualBox. For write operation also except for some small block size write operations in VMware shows better performance than in VirtualBox.

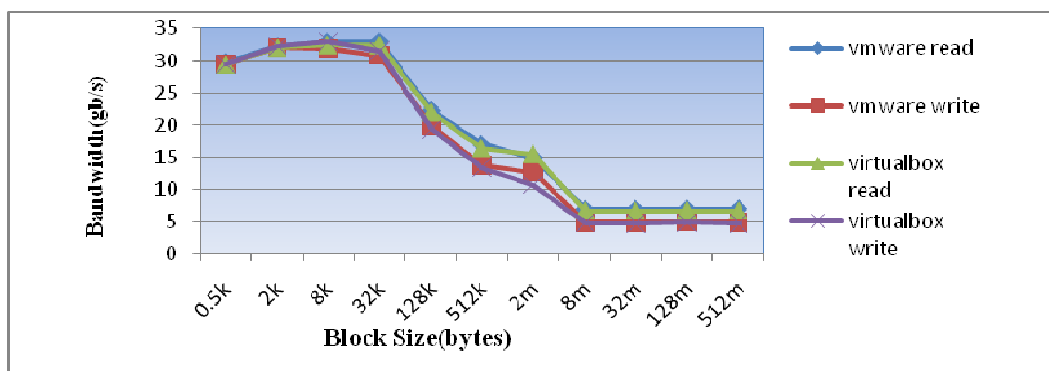


Fig.1: Memory bandwidth of read and write in LMbench

Comparisons of other system operations are shown in Tab.1. The interprocess communication via TCP/IP is measured by passing a token back and forth between two processes (lat_tcp), context switching is also included. VMware showed a better performance than VirtualBox. In the following experiments VirtualBox outperformed VMware, we measured interprocess connection latencies, which is the time for creation and connection of an AF_INET socket to a remote server. The time to open and close a file, i.e. syscall. Measured the interprocess latency through pipes, here we used two process communicating through UNIX pipe and passes a token back and forth. Measured fork and exec. 'fork' is the time it takes to split a process into 2 identical copies and have one exit, 'exec' is the time it takes to create a new process and have that new process run a new program. Then Context Switch times have been measured by giving size of the process and number of processes as input. The results are shown in the Tab.2.

Table. 1: System operation times

	VMware(μ s)	VirtualBox(μ s)
lat_tcp	1.7625	.6127
connect	25.78575	91.436
syscall	3.45395	2.217
lat_pipe	24.59705	177.7406
fork	497.4091	2948.6757
exec	524.0909	3225.7659

Table. 2: Context switch times

	VMware(μ s)	VirtualBox(μ s)
2p0k	9.46	66.86
2p16k	11.61	75.52
2p64k	11.76	75.05
8p16k	12.12	83.93
8p64k	11.98	71.90
16p16k	12.25	79.36
16p64k	12.65	76.27

Tab.3 and Tab.4 shows the result of file system create/delete performance and we can conclude that the creations per second and deletions per second for file sizes 0k, 1k, 4k, 10k is more for VMware than VirtualBox. The time for data movement through pipes is measured by creating a UNIX pipe between 2 processes. The message is passed in 64KB chunks. The results are shown in Tab.5, where VMware outperforms VirtualBox.

Table. 3: VMware File system create/delete results

Size of the file	Number of files created	Creations per second	Removal per second
0k	339	62639	106269
1k	194	34282	82499
4k	130	22891	82569
10k	71	12147	66508

Table. 4: VirtualBox File system create/delete results

Size of the file	Number of files created	Creations per second	Removal per second
0k	82148	58896	103235
1k	54659	13454	48254
4k	45129	19493	50692
10k	31206	12497	46094

Table. 5: Data movement through pipes

Total Bytes	VMware(MB/s)	VirtualBox(MB/s)
30 M	1528.41	152.16
100M	1060.59	149.59

To investigate the virtualization of file systems I/O, we separately run IOzone on virtualized Ubuntu, virtualized XP in VMware and on virtualized Ubuntu and virtualized XP in VirtualBox. Here we have

checked the read, reread, write and rewrite performance of different virtual machines. We measured the read, reread, write and rewrite performances by fixing the record size to 1MB and varying the file size and the output will be bandwidth in gb/s. The results of read, reread performance are shown in Fig.2 and the results of write and rewrite performance are shown in Fig.3.

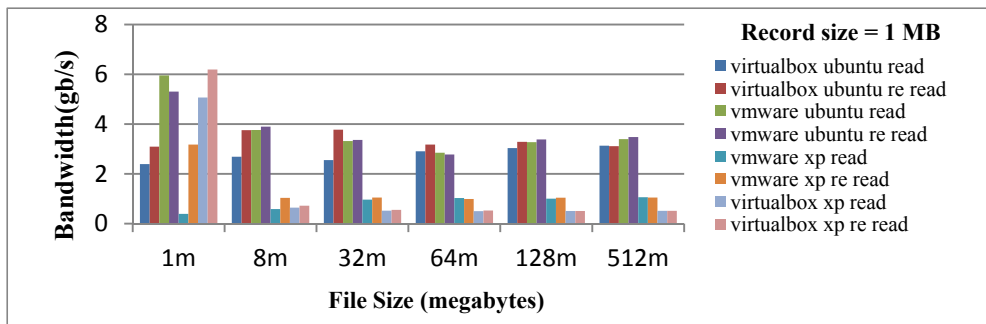


Fig. 2: Read and Re read bandwidth of different environments and cases in IOzone

Fig. 2 shows that except for small file sizes VMware outperforms VirtualBox. Since the reread and rewrite performance are better for VMware it indicates that the buffer utilization is efficient here.

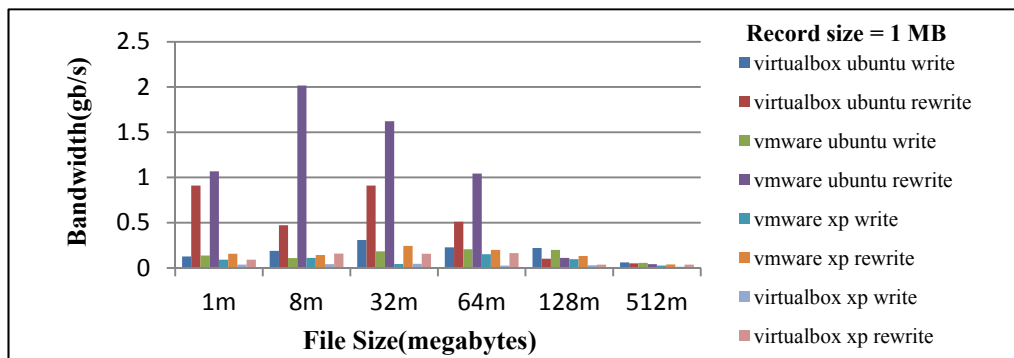


Fig. 3: Write and Re write bandwidth of different environments and cases in IOzone

6. Conclusion and Future Works

Firstly we described the basic knowledge about VMware and VirtualBox and two benchmarks LMBench, IOzone and conducted experiments to measure the performance of virtualized Ubuntu, XP on VMware and VirtualBox. We took a glance at the overhead of CPU, memory and I/O operations and revealed some interesting hints. In future we will measure the network performance as well as include more virtual machine monitors for comparison and also include some more benchmarks.

7. Acknowledgements

I wish to express my deep sense of gratitude to Prof. Biju R Mohan, Prof. Dinesh Naik and my classmate Vasudevan M.S, Department of Information Technology, National Institute of Technology Karnataka for giving support in doing this project.

8. References

- [1] Jianhu Che, Qinming He, Qinghua Gao and Dawei Huang, "Performance Measuring and Comparing of Virtual Machine Monitors," IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008.
- [2] Xianghua Xu, Feng Zhou, Jian Wan Yucheng Jiang, "Quantifying Performance Properties of Virtual Machine," International Symposium on Information Science and Engineering, 2008.
- [3] J.E.Smith and Ravi Nair, "An Overview of Virtual Machine Architectures".
- [4] Jianhua Che and Yong Yu, "A Synthetical Performance Evaluation of OpenVZ, Xen and KVM," IEEE Asia-Pacific Services Computing Conference, 2010.
- [5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, "Xen and

- the art of virtualization*,”*ACM SIGOPS Operating Systems Review*, 37(5):164-177, 2003.
- [6] T. Deshane, Z. Shepherd, J. Matthews, M. Ben-Yehuda, A. Shah and B. Rao,”*Quantitative Comparison of Xen and KVM*,” *Xen Summit*., June 23-24, 2008.
- [7] P. Padala, X. Zhu, Z. Wang, S. Singhal and K. Shin,”*Performance Evaluation of Virtualization Technologies for Server Consolidation*,”*Technical Report HPL-2007-59*, HP Labs, April 2007.
- [8] V. Inc. A Performance Comparison of Hypervisors. Technical report,VMWare Inc., 2007.
- [9] V. Inc. Understanding Full Virtualization, Paravirtualization and Hardware Assist. Technical report, VMWare Inc., 2007.
- [10]B. Clark, T. Deshane, E. Dow, S. Evanchik, M. Finlayson, J. Herne and J. Matthews, “*Xen and the Art of Repeated Research*,” In *Proceedings of the 2004 USENIX Annual Technical Conference*, pages 135-144, 2004.
- [11]P. Apparao, R. Iyer, X. Zhang, D. Newell and T. Adelmeyer, “Characterization & analysis of a server consolidation benchmark,” In *Proceedings of the 4th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments(VEE’08)*, USA: ACM New York, pages 21-30, 2008.
- [12]A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, W. Zwaenepoel, ”Diagnosing Performance Overheads in the Xen Virtual Machine Environment,”*Proc. of the 1st ACM/USENIX International Conference on Virtual Execution Environments(VEE)*,13–23, 2005.
- [13]P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield,” *Xen and the Art of Virtualization*,” *Proc. of the nineteenth ACM Symposium on Operating Systems Principles(SOSP)*, 164–177, 2003.
- [14]L. McVoy and C. Staelin,” *LMbench: Portable Tools for Performance Analysis*,” *Proc. of the 1996 AnnualConference on USENIX Annual Technical Conference*, San Diego, CA, 279–294, 1996.
- [15]VMware Inc. Understanding Full Virtualization, Paravirtualization and Hardware Assist. White paper.November 11, 2007.
- [16]M. Rosenblum and T. Garfinkel,”*Virtual Machine Monitors: Current Technology and Future Trends*,”*IEEE Computer*, 38(5):39–47, 2005.
- [17]<http://www.vmware.com/>
- [18]<https://www.virtualbox.org/>
- [19]<http://www.bitmover.com/lmbench/>
- [20]D. Capps and W.D. Norcott. IOzone Filesystem Benchmark. <http://www.IOzone.org>. 2004.