International Journal of Scientific & Engineering Research, Volume 7, Issue 11, November-2016
ISSN 2229-5518

Evaluation of different Hypervisors Performances using Different Benchmarks

Shrutika Dhargave
Computer Enginnering MIT, Pune
shrutika.dhargawe@gmail.com

Prof. S. C. Karande
Computer Enginnering MIT, Pune
shridevi.karande@mitpune.edu.in

Abstract: Virtualization has become a popular way to make more efficient use of server resources within both private data centers and public cloud platforms. Hypervisors are widely used in cloud environments and their impact on application performance has been a topic of significant research and practical interest. While recent advances in CPU architectures and new virtualization techniques have reduced the performance cost of using virtualization, overheads still exist, particularly when multiple virtual machines are competing for resources. The paper will cover the comparisons of some hypervisors based on their performance. The hypervisors which are to be compared are XEN, VMware, KVM, and Hyper-V. Thus, the paper gives a brief idea about each hypervisors and theirs performances at different levels.

Keywords: Virtualization, Hypervisor, Performance, Benchmarks

I. Introduction

The recent growth in cloud environments has accelerated the advancement of virtualization through hypervisors; however, with so many different virtualization technologies, it is difficult to ascertain how different hypervisors impact application performance and whether the same performance can be achieved for each hypervisor. Many different hypervisors (both open source and commercial) exist today, each with their own advantages and disadvantages.

II. Background Knowledge

1. XEN server:

Xen is an open source hypervisor originally developed at the University of Cambridge and now distributed by Citrix Systems, Inc. The first public release of Xen occurred in 2003 [1]. It is designed for various hardware platforms, especially x86, and supports a wide range of guest operating systems, including Windows, Linux, Solaris and versions of the BSD family. Xen employs para-virtualization from the very beginning. Through para-virtualization, Xen can achieve very high performance, but it has the disadvantage of supporting Linux only; and that Linux has to have a modified kernel and bootloader, and a fixed layout with two partitions, one for hard disk and one for swap. Xen also implements support for hardware-assisted virtualization. In this configuration, it does not require modifying the guest OS, which make it possible to host Windows guests.

2. VMware ESXi:

VMware ESXi is an operating system-independent hypervisor based on the VM kernel operating system interfacing with agents that run atop it. ESXi is the exclusive hypervisor for VMware vSphere 5.x licenses. VMware describes an ESXi system as similar to a stateless compute node. State information can be uploaded from a saved configuration file. ESXi's VM kernel interfaces directly with VMware agents and approved third-party modules. Virtualization administrators can configure VMware ESXi through its console or the VMware vSphere Client and check VMware's Hardware Compatibility List for approved, supported hardware on which to install ESXi [2].

3. KVM:

KVM is a hardware-assisted virtualization developed by Qumranet, Inc. and was merged with upstream mainline Linux kernel in 2007, giving the Linux kernel native virtualization capabilities. KVM make use of the virtualization extensions Intel VT-x and AMD-V. In 2008, Red Hat, Inc. acquired Qumranet. KVM is a kernel module to the Linux kernel, which provides the core virtualization infrastructure and turns a Linux host into a hypervisor. Scheduling of processes and memory is handled through the kernel itself. Device emulation is handle by a modified version of QEMU [3]. The guest is actually executed in the user space of the host and it looks like a regular process to the underlying host kernel. KVM supports I/O para-virtualization using virtio subsystem. Virtio is a virtualization standard for device (network, disk, etc.) drivers where the guest's device driver is aware of running in a virtual environment, and communicates directly with the hypervisor. This enables the guests to get high performance network and disk operations.

4. Hyper-V:

Microsoft could not ignore the virtualization trend. Microsoft introduced Hyper-V as a virtualization platform in 2008, and it continued to release new Hyper-V versions with new Windows server versions. So far, there are a total of four versions, including Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2 and Windows Server 2008.Since Hyper-V's debut, it has always been a Windows Server feature, which could be installed whenever a server administrator decided to do so. It's also available as a separate product called Microsoft Hyper-V Server. Basically, Microsoft Hyper-V Server is a standalone and shortened version of Windows Server where Microsoft cut out everything irrelevant to virtualization, services and Graphical User Interface (GUI) to make the server as small as possible. Plus, without the bells and whistles, the server requires less maintenance time and it is less vulnerable, because, for example, fewer components mean less patching. Hyper-V is a hybrid hypervisor, which is installed from OS (via Windows wizard of adding roles) [4].

III. Performance based on different methods

1. Hadoop Benchmark:

In paper [5], Hadoop implementation of the MapReduce framework is used. The Hadoop cluster comprises of 4 nodes that all reside in the same physical machine. Each node has one virtual core pinned to a different physical core, allotted 2 GB of memory, 50 GB of disk space, and is set to run at most 2 map tasks or 2 reduce tasks. We run 3 different Hadoop benchmarks found in version 1.0.2 including TestDFSIO Write/Read, and TeraSort and 3 benchmarks within the HiBench suite, namely, Word Count, K-Means Clustering, and Hivebench. Each of these benchmarks was run ten times and the average was used

For HiBench, he performance impact when using different hypervisors was negligible. Performance difference between the hypervisors for each benchmark is not very significant with the highest percentage difference being 15.7% for Hive Aggregate. They shows very high saturation of the CPU for each hypervisor for word count. The disk is being utilized at an average of less than 40% for each hypervisor.

For TestDFSIO Write on a 4 node Hadoop cluster for each hypervisor using varied write size to HDFS from 5GB to 25GB with 5GB increments shows that KVM has the slowest write throughput for all sizes at an average of 42.0 MB/s. The performance variation is due to the efficiency of request merging when processing a large number of write requests to disk. For the read-intensive benchmark TestDFSIO Read, KVM reads data at an average throughput of 23.2 MB/s for the four data sizes. If we look at the physical core utilizations for these VCPUs at the host level only KVM is heavily utilizing the physical CPUs.

TeraSort is both a CPU and disk intensive benchmark. Each worker has to read data from disk, perform the sort (where intermediate data gets written back to the disk), sends it to the reducer node where the reducer again performs an aggregate sort and writes the final result back to disk. For a combination of heavy read and write-intensive I/O and CPU used for sorting, Xen performed the best and was able to sort the data the quickest while KVM's performance was slightly worse than Xen's performance. For TestDFSIO Read, KVM was faster than Xen, but in this map phase Xen is faster than KVM. This difference is due to the added CPU needed to compute the sort on the data; whereas Xen can use dom0 to offload some of the disk read which frees up CPU for the VMs in Xen to perform the sort, the KVM VMs must use its own CPU to read and perform the sort. During the reduce phase, we see that Xen completes faster while KVM takes the longest time. In TestDFSIO Write, Xen was found to be slower than KVM.

2. GPU Pass-through Performance:

GPU virtualization and GPU-pass-through are used within a variety of contexts, from high performance computing to virtual desktop infrastructure. Accessing one or more GPUs within a virtual machine is typically accomplished by one of two strategies: 1) via API remoting with device emulation; or 2) using PCI pass-through. We characterize GPGPU performance within virtual machines across two hardware systems, 4 hypervisors, and 3 application sets [6].

KVM again performs well across both the Delta and Bespin systems. In the case of the Delta system, in fact, KVM, significantly outperforms the base system. VMWare perform close to the base system, while Xen achieves between 72–90% of the base system's performance. LAMMPS is unique among our benchmarks, in that it exercises both the GPU and multiple CPU cores. LAMMPS performs well across both hypervisors and systems. Surprisingly, LAMMPS showed better efficiency on the Delta system than the Bespin system, achieving greater than 98% efficiency across the board, while Xen on the Bespin system occasionally drops as low as 96.5% efficiency. LULESH is a highly compute-intensive simulation, with limited data movement between the host/virtual machine and the GPU, making it ideal for GPU acceleration. Overall, we see very little overhead, there is a slight scaling effect that is most apparent in the case of the Xen hypervisor.

3. SIGAR Framework:

SIGAR Framework for Xen-In CPU utilization test, lower CPU consumption and less variation (in case of medium and high workloads) is better for a guest OS. In case of Memory test, high available memory indicates superior performance of a guest OS. In case of Disk I/O tests higher sequential read and write are the signs of better guest OS on the XenServer Hypervisor. And in network performance high transfer rate and less variation (in the case of medium to high workloads) indicates the better performance for a guest OS[7].

SIGAR Framework for ESXi- In CPU utilization test, lower CPU consumption and less variation (in case of medium and high workloads) is good for a guest OS. In case of Memory test, high available memory indicates superior performance of a guest OS. In case of Disk I/O tests higher read and write are the signs of better guest OS on the ESXi Hypervisor. And in Network performance high transfer rate and less variation (in the case of medium to high workloads) indicates the better performance for a guest OS[8].

4. FTP and HTTP approach:

Regarding to Transfer Time, the real environment has a performance approximately 2 times better than other hypervisors. It means that the Hypervisor includes a consider time delay during FTP transmission. The real environment has a better performance in CPU consumption and Memory Utilization than the Hypervisors. The Time Transfer in OpenVZ_FTP Server is smaller than that of the other hypervisors. This happens because OpenVZ hypervisor is less complex than the others. In OpenVZ each Guest OS has the same kernel as Host OS and each Guest looks like a simple process.

XEN-PV_FTP Server has a better performance in CPU consumption, because XEN-PV has a better isolation between Guest OS and Host OS. The Memory Utilization in XEN-PV_FTP Server is better than the memory utilization in other Hypervisors. The second one is OpenVZ Hypervisor. The worse Memory Utilization is KVM_FTP/Web Server. This is because KVM offers more overhead than other Hypervisors. The Transfer Time in OpenVZ and XEN-PV Web Server is better than that of other hypervisors. CPU consumption in KVM–PV is better than in KVM-FV because of the implementation of "virtio driver" in KVM-PV [9].

IV. Conclusion and Future Scope

The comparisons of the hypervisors are only based on performances. Depending on the framework, each hypervisor has good performances with a slight difference from the other hypervisors. It is very difficult to conclude that which hypervisor is better among them. Thus depending on the project criteria appropriate hypervisor can be selected. As stated before, hypervisors are compared based on performances. The hypervisors can also be compared on the base of ease of development for future work.

References

[1] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt and Andrew Warfield, "Xen and the Art of Virtualization," Proceedings of the ACM Symposium on Operating Systems Principles, 2003.

[2]http://searchvmware.techtarget.com/definition/VMware-ESXi.

[3] A. Singh, "An Introduction to Virtualization", kernelthread.com, January 2004.

[4] https://hyperv.veeam.com/blog/what-is-hyper-v-technology.

[5] Jack Li, Qingyang Wang, Deepal Jayasinghe, Junhee Park, Tao Zhu, Calton Pu, "Performance Overhead Among Three Hypervisors: An Experimental Study using

Hadoop Benchmarks", IEEE International Congress on Big Data (BigData Congress) 2013.
[6] John Paul Walters, Andrew J. Younge, Dong-In Kang, Ke-Thia Yao, Mikyung Kang, Stephen P. Crago, and Geoffrey C. Fox, "GPU Pass-through Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications" IEEE 7th International Conference on Cloud Computing (CLOUD), 2014.
[7] Vijaya Vardhan Reddy, Dr. Lakshmi Rajamani, "Performance Evaluation of Operating Systems in the Private Cloud with XenServer Hypervisor Using SIGAR Framework",9th International Conference on Computer Science & Education (ICCSE), 2014.
[8] Vijaya Vardhan Reddy, Dr. Lakshmi Rajamani, "Evaluation of Different Operating Systems Performance in the Private Cloud with ESXi Hypervisor using SIGAR Framework", 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), 2014.
[9] Igli Tafa, Ermal Beqiri, Hakik Paci, Elinda Kajo, Aleksandër Xhuvani, "The evaluation of Transfer Time, CPU Consumption and Memory Utilization in XEN-PV, XEN-HVM, OpenVZ, KVM-FV and KVM-PV Hypervisors using FTP and HTTP approaches", Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2011