

Testing Various Virtualization Solution for Different Cloud Workloads

Ghanshyam ji jha¹, Prem Prakash Agrawal², Dr. D. Nageswara Rao³

¹Student, School of Computer Science and Engineering, Galgotias University, Greater Noida, India

²Professor, School of Computer Science and Engineering, Galgotias University, Greater Noida, India

³Professor, School of Computer Science and Engineering, Galgotias University, Greater Noida, India

Abstract - In early days when real dedicated servers were, used people did not cared for top performance out of the machine as physical machines in most cases there were always some headroom left for any burst use. Since technologies like virtualization came, our focus changed to getting most performance out for machines for its use cases. This project is designed to testing various virtualization solutions for different cloud workloads and technology such as qemu, lxc, hyperv for various cloud consumption like hosting, high performance computing, database hosting and various other use, so we can give a guideline regarding the use of hypervisor software to get most performance out of a machine.

Key Words: Virtualization, Hypervisors, Cloud Computing, Web Hosting.

1. INTRODUCTION

Virtualization is the process or method to create an abstract layer of physical resources such as CPU, Network as well as Storage. This abstraction allows for better utilisation of the hardware. Virtualization came into existence in 1960s and was brought into practice by IBM first. They wanted to utilize their existing mainframe resources by making them virtually available as they were being wasted.

Many years ago, we started using virtualization technology to test servers and use resources more efficiently. Virtualization was not very popular when VMware came on the market as a hypervisor. During the 1980's and 1990's, Desktop Computing came into existence along with x86 servers. This is why Virtualization was eventually dropped. Desktop computing and x86 servers. This is why Virtualization was eventually forgotten. The introduction of client-server applications and wide popularity of Windows along with Linux OS made server computing expensive. Many challenges lie ahead, and not enough disaster prevention that leads to rise of Virtualization of x86 again. Virtualization improves scalability and resource utilization by a high degree. Using virtualization, several operating systems can run simultaneously on a single CPU. This reduces overall costs and differs from multitasking, which means running several programs on the same OS. Virtualization is a major enabler for Cloud Computing.

However, even Virtualization is not perfect. There are many emerging technologies, which have proved to improve virtualization. Will these technologies replace virtualization in the future? This is one of the answers we are looking for. One of the technologies that is making its way into the virtualization world is Cloud Containerization. Containerization is an easy way to complete machine virtualization, which involves access application in its equipment. Your operating environment. It offers many benefits of loading applications on a virtual machine, as the application can be run on any physical machine without having to worry about dependencies.

Another major breakthrough technology is Server less Computing. When using an unstructured computer, developers write the code, and the cloud server monitors the rest. The developer should not think of servers, operating systems, software, or management. Of course, there is a physical server to runs the code, but that is the cloud service provider's job. As with the micro service/container scenario, server less computing bypasses the virtual machine layer and functions run on bare metal. At this point, server less computing is relatively naive and use cases are in short supply..

2. BACKGROUND & RELATED WORK

The data center industry has gone far beyond simple virtualization, exploring new ways to make virtualization an even stronger platform. Virtualization has become a trusted workhorse in IT environments of all sizes. This allows organizations to maximize the utilization of their hardware assets, reduce power consumption and move one system to another without disruption to a few services. Large organizations are adopting virtualization nowadays. Not many people know about it, but it is

also accessible an important role in cloud computing. Because of virtualization, the cloud can effectively provide the scalable, scalable, affordable services they are so well known for.

Virtualization is now being replaced by several new technologies, which will be much more popular in the future. This thorough study on Virtualization along with the experimental research is going to give a deeper understanding of how these technologies work as well as a comparison among them.

2.1 Memory Management in VMWare ESXi:

Let's assume that your Suppose your ESXi host server has a total available memory as 50 GB. You have 10 virtual machines, which are configured with 4 GB each. So that's a total of 50GB. But you might have noticed that the actual amount of memory available is 40GB. This can be achieved using VM memory management techniques. Not all VMs use 100% of allocated memory at all times. If this happens, the host will use some memory restoration techniques on its own to deal with this condition effectively.

A few of the techniques, which are widely used to manage memory, are as follows:

1. TPS: Transparent Page Sharing
2. MB: Memory ballooning
3. Memory Compression
4. Hypervisor-level memory swapping

An ESXi host never tries to restore memory through memory management

Techniques until it is contested in memory. Techniques for restoring such memory

MB- Memory ballooning or MC- Memory Compression or MS- Memory Swapping Itake action based on the available free host memory.

2.2 Below are the 4 states of ESXi:

2.2.1 High: By default, transparent page sharing will always work.

2.2.2 Soft: MB launches when our ESXi server enters soft state, and it will stay active until ESXi returns to a high state.

2.2.3 Hard & Low:

Hard and low: ESXi uses memory compression and Hypervisor-level memory swapping when ESX is in hard or low mode.

2.2.4 Low:

If the memory usage is more than low, the host stops creating more pages for VMs. In addition, it will continue to compress and replace until it clears more memory.

Before we had vSphere 5, High state was by default 6% whereas; we had Soft state at 4%.

Along with this, Hard State was at 2% with Low State at only 1%. If the free memory is anywhere less than given percentage, the host puts to use proper memory restoration techniques in order to restore all of the memory. Actually, ESXi will contain not just one memory. It is not necessary to protect it.

For example, the ESXi 5 host can run using 2 Terabytes of memory. With these defined values, like vSphere 5.0, a host will begin to require memory even if it has as much as 100GB of memory left. This is a bad idea. In vSphere 5.x, these values have been modified in order to handle host restoration techniques for the ESXi. In vSphere, the high state level is changed based on the amount of memory the host possesses.

Soft -> (64% of High)

Hard -> (32 % of High)

Low -> (16 % of High)

Host Memory	HIGH (900 MB for 28 GB + 1 % of all memory above 28 GB)	SOFT (64 % of High)	HARD (32% of High)	LOW (16% of High)
32 GB	941 MB	603 MB	302 MB	151 MB
48 GB	1,105 MB	707 MB	354 MB	177 MB
64 GB	1,269 MB	812 MB	406 Mb	203 MB
96 GB	1,596 MB	1,022 MB	511 MB	255 MB
128 GB	1,924 MB	1,231 MB	616 MB	308 MB
512 GB	5,856 MB	3,748 MB	1,874 MB	937 MB
1 TB	10,199 MB	6,528 MB	3,264 MB	1,631 MB
2 TB	20,685 MB	13,239 MB	6,620 MB	3,309 MB

Table 1: Above are some examples of Memory Reclamation Levels in ESXi.

3. METHODOLOGY

The methodology for our performance comparison of hypervisors is to testing various virtualization solutions for different cloud workloads and technology such as qemu, lxc, hyper v for various cloud consumption like hosting, high performance computing, database hosting and various other use.

3.1 Experimental Setup: The purpose of our system setup is to provide complete fairness in all aspects that can affect system performance.

3.2 Hardware Setup: For fair comparison, hardware settings are completely identical for all supervisors using a single server machine (Dell R620 server), which has two processor 2* Intel xeon2670 (20 core) with RAM 128GB DDR3 and HDD 3*900GB SAS 10k.

3.3 Tools:

3.3.1 Datadog: is a cloud-based service that provides monitoring of servers, digital, tools and services through a SaaS-based data analytics platform. we are using Datadog to The analysis includes CPU utilization, Disk Usage, Disk Latency, Memory Breakdown and MySQL and Apache statistics. The load is at its peak from 4:00pm to 7:00pm.

3.3.2 Sysbench is an open source bench-marking tool. It examines CPU usage related to the operating system, memory usage, Disk IO and MySQL performance. It is particularly important to use these computer parameters when starting a critical database.

3.4 Hypervisors we are using:

3.4.1 KVM: Kvm represents the latest generation of open source virtualization. The aim of the project was to create a modern hypervisor who builds on the experience of the previous generation. It simply makes the kernel into hypervisor when you install the kvm kernel module. Now, because the Linux kernel is the supervisor, it can enjoy the modifications to the standard kernel along with memory support as well as scheduler.

3.4.2 HYPER-V : Microsoft Hyper-V is a virtualization technology for x64 versions of Windows Server. It exists in two versions: as a standalone product called Hyper-V Server and as a installable role / component in Windows Server [18]. There is no difference between MS Hyper-V in either of these two versions. The hypervisor is the same regardless of the installed version.

3.4.3 HYPER CORE: The uniqueness of the scale computing solution is the patented Hyper Core software. Hyper Core preloaded on all HC3 nodes, ready to deploy straight out of the box, well-tuned for any type of HC3 node without additional licensing or installations of Hyper Core, Hyper core always monitors all virtual machines, software, and hardware to automatically detect and respond to detect and respond to daily events, maintain application availability, and simplify data center management. Scale Computing HC3 and Hyper Core architecture have been designed to provide high quality, high-quality computing and storage services while maintaining ease of use and risk through smart software automation and architecture simplicity.

3.4.4 LXC: Linux Containers (LXC) is a project aimed at building a toolkit for isolating processes from other processes. As mentioned, the cone is not developed as a security feature and can be escaped from the system - LXC tries to create environments (containers) that should be immune to its process and child processes and protect the system even if an attacker manages to escape the container. Besides, it provides an interface for reducing the resource consumption of delicate containers.

3.4.5 QEMU: QEMU is an open source machine virtualization emulator. Run programs for a different Linux / BSD purpose, in every supported architecture Run operating systems for every computer, in every supported architecture: It mimics the machine processor through dynamic binary translation and provides a set of different hardware models and devices for the machine, enabling it to run a variety of guest operating systems. It can also be used with KVM to run virtual machines at near-natural speed (by utilizing hardware extensions such as Intel VT-x). QEMU can also emulate user-level processes, enabling complex applications for a single architecture.

3.4.6 ESXI: - VMware ESXi (formerly ESX) VMware vSphere is a software package that includes many software components such as vCenter, ESXi and the vSphere client [13]. VSphere is not a specific software that you can install and use, "it's just a package name that has other subcomponents." The hypervisor comes in the form of VMware ESXi, which is a type 1 hypervisor (bare metal). All virtual machines or guest operating systems are installed on the ESXi server. Different part of a vSphere suit, vSphere or vCenter client is needed to install, manage, and access the virtual servers located above the ESXi server.

3.5 Tests performed to check for more effective Containerization:

Actual dataset for creating a new instance (Total size in GB vs. Number of Instances):

CentOS installation size:

Initial Installation size: 1.3 GB
After critical update: 2.0 GB

OpenVZ sizing:

Initial installation size: 3.0 GB
After CentOS template: 3.2 GB

	A	B	C	D	E	F	G	H	I	J
		1	2	3	4	5	6	7	8	9
Typical Virtualization		2	4	6	8	10	12	14	16	18
OpenVZ		5.4	6.2	7	7.9	8.7	9.5	10.3	11.1	11.9

Table 2: Dataset for creating a new instance.

K	L	M	N	O	P	Q	R	S
10	20	30	40	50	60	70	80	90
20	40	60	80	100	120	140	160	180
12.7	20.5	38.3	36.1	43.95	51.7	59.5	67.3	75.2

Table 3: Dataset for creating a new instance (contd.)

3.6 For High Performance Computing testing:

we are computing to "Initializing random number generator, limit to Prime numbers 200000" in 6 different hypervisors on same hardware one at a time to figure out which hypervisor is working more efficiently using sysbench to Benchmark Cpu performance.

4. RESULT

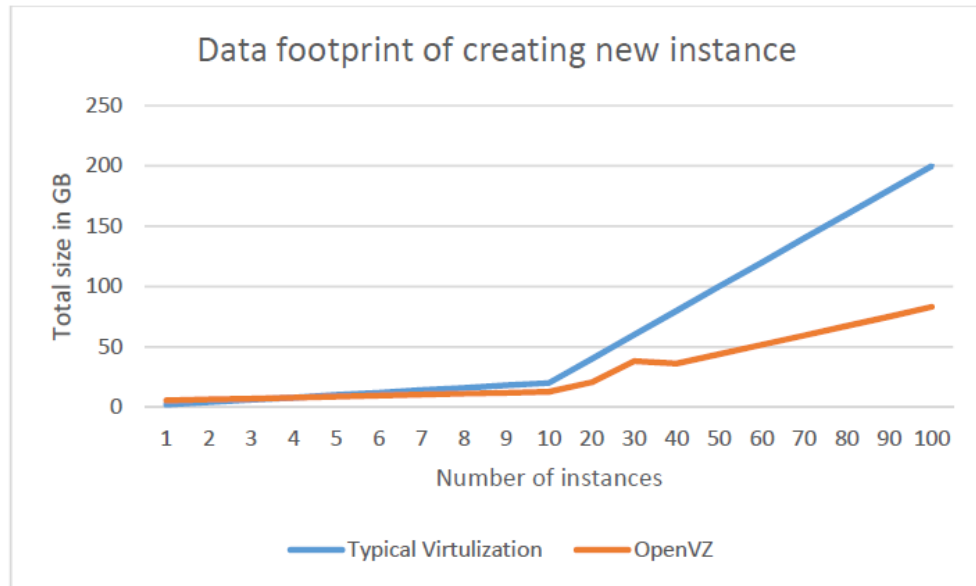


Chart 1: Data footprint of creating a new instance.

4.1 SYSBENCH RESULT

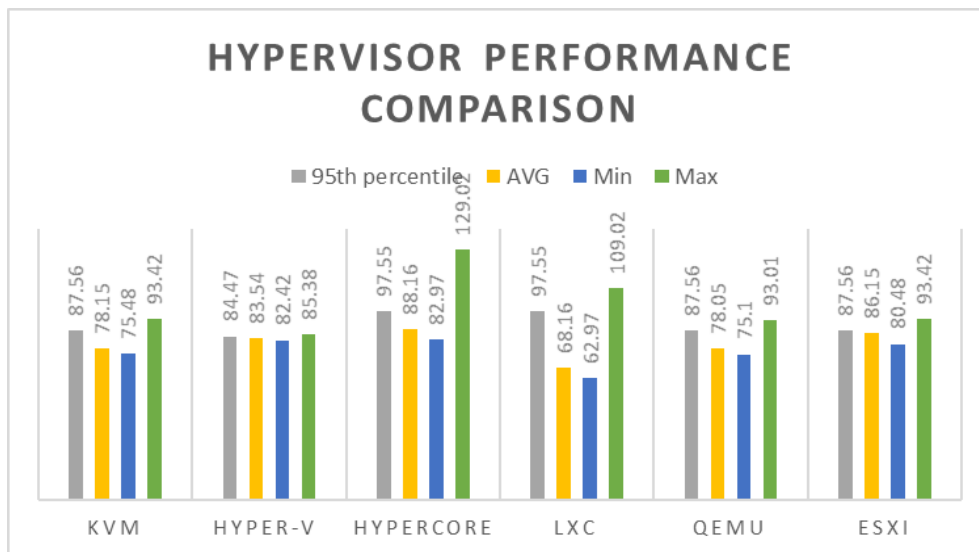


Chart 2: performance comparison chart.

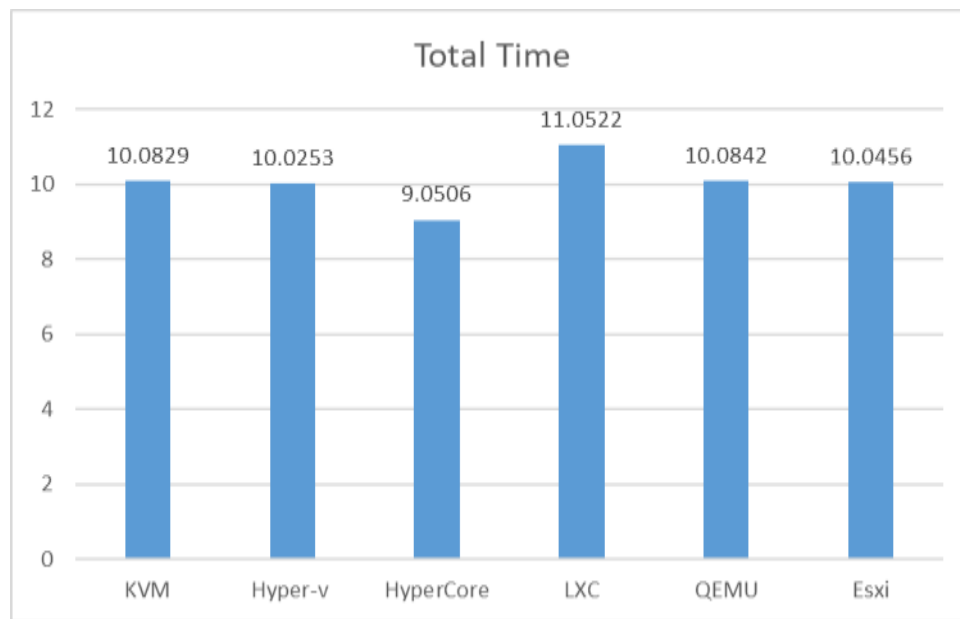


Chart 3: maximum time taken to complete the computing.

5. CONCLUSIONS

As we can see from the above graph in Chart 1, the size for creation of a new instance is almost the same for traditional VM as well as OpenVZ at first. But after a while, when the number of instances increase significantly (in our case after 10 instances), the size of virtual instance increases too heavily compared to OpenVZ. This proves that for large number of instances and better utilization, Virtualization may not be the best choice. It could incur significant overhead when it comes to size. Containerization is a lightweight alternative to Virtualization.

Specific containers, code, applications or processes isolated. This gives everything in the container a neat envelope to manage, including moving it across different hosts. While you might think of a virtual machine deploying a server for multiple operating systems, containers run above the operating system, so unlike a VM, they do not require an operating system to boot when created. In essence, they can virtualize an operating system to provide an easier package of an app than a VM. We have so far divided a little into the advantages of using containers.

After a thorough analysis of the six hypervisors, we came across some interesting results as shown in the graph above. HC3 Hypercore took the least amount of time (09.0506s) to generate random numbers up to the provided limit which was 2 00,000. We can conclude that it is the fastest. Second is Hyper-V with 10.0253s, followed by ESXi, KVM, Qemu and finally the slowest LXC.

In terms of performance latency, the following are the best and worst respectively:

LXC has the least minimum latency among all, while HC3 Hypercore has the greatest minimum latency among all. In terms of average, LXC has the least average while HC3 Hypercore has the most. Hyper-V beat LXC in terms of least maximum latency, while HC3 Hypercore has the maximum. At last, Hyper-V has the minimum 95th percentile while HC3 Hypercore and LXC have the maximum.

We can conclude that LXC followed by have the best performance in terms of latency. HC3 Hypercore while being the fastest of them all, has the maximum latency.

Hyper-V and ESXi are easy to use and install but it comes under Premium pricing, where as KVM and Qemu is open source but it is quite difficult to install. For web/cloud Hosting we do not need computational power, what we need a system like Lxc which does not need a VM dividing servers into more than one operating system. In addition, an LXC provides Virtualization of an operating system, which is so much lighter as compared to a Virtual Machine. HC3 Hypercore is a Premium pricing but it is quite optimized to do high power computing. In the end, it is a matter of preference. All major cloud providers use different kinds of hypervisors based on their needs.

ACKNOWLEDGEMENT

This research was partially supported by Mr. Amit Chaudhary 'Amit2cha@gmail.com'. We thank our colleague from "The Roasters" who provided insight and licensing of all products. This paper and the research behind it would not have been possible without the exceptional support of my supervisor, Mr. Prem Prakash Agrawal. His enthusiasm, curiosity and willingness to focus on everything motivated me and continued my work. I want to thank my parents, without their blessings, I cannot fulfill my purpose.

REFERENCES

- [1.] "History of Virtualization" by harrisdy (October 19, 2009)
http://www.infobarrel.com/History_of_Virtualization
- [2.] "Will containers kill the virtual machine?" by Brandon Butler (APR 15, 2015)
<https://www.networkworld.com/article/2910559/will-containers-kill-the-virtual-machine.html>
- [3.] Containerization by Kiante Evans
<https://quizlet.com/322447201/containerization-flash-cards/>
- [4.] "What's the future of server virtualization" by Neal Weinberg (July 11, 2018)
<https://www.networkworld.com/article/3285906/whats-the-future-of-servervirtualization.Html>
- [5.] "Virtualization: A look at the past, present and future of IT's trusty workhorse"
by Contel Bradford (May 29, 2019)
<https://blog.storagecraft.com/virtualization-a-look-at-the-past-present-and-future-of-its-trusty-workhorse/>
- [6.] "VMWare memory management part 1- Understanding ESXi host memory states" by Mohammed Raffic (December 22, 2014)
<http://www.vmwarearena.com/esxi-memory-management-part-1-understanding-esxi-host-memory-states/>