2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)

Comparison between common virtualization solutions: VMware Workstation, Hyper-V and Docker

Ziyu Li*
School of Advanced Technology,
Xi`an-Jiaotong-liverpool University,
Suzhou, China
ziyu.li18@student.xjtlu.edu.cn

Abstract: **This work aims to present the variations between three common virtualization platforms: VMware Workstation, Hyper-V and Docker in the respect of hardware efficiency with different operating systems (Ubuntu20.04 and Windows 10). Firstly, the differences in the theoretical architecture of these three virtualization solutions are discussed in order to provide a brief reference for the testing of operational efficiency in the following part. Subsequently, some common test applications (sysbench, fio, mbw, Fritz Chess Benchmark and Crystal Disk Mark) were run to test their efficiency in CPU, hard disk and memory. Eventually, these statistics are collected in the form of graphs for comparison. The results of the comparison clearly reveal the advantages of VMware workstation in the efficiency of virtual machine hardware operation, while Hyper-V, supported by Microsoft, is inferior to the former in both ubuntu and Windows 10 environments. Meanwhile, Docker, as a representative of an emerging container virtualization technology, has demonstrated a superior performance closer to physical machine performance than traditional virtual machine technologies because of its unique architecture.**

Keywords: Virtualization, Comparison, Container, Virtual machine.

I. Introduction

Virtualization technology, which lies at the root of the age of cloud computing, is widely applied in the construction of cloud systems, enabling higher utilization of servers and better isolation of user-spaces. Nowadays, with the development of virtualization for decades, there is a great quantity of virtualization techniques, the application of which mainly depends on specific user requirements: some approaches of virtualization focus on high hardware utilization rate of the host, whereas other approaches aim at greater compatibility. These diverse strengths of different techniques, to a great extent, is determined by variation of architecture. From a broad perspective, the most widely used virtualization techniques in recent years can be classified into the virtual machine technology and the container technology, according to their underlying structures.

The thriving of modern virtual machines can date back to the late 1990s, when the Disco research project team from Stanford University was devoted to the resurgence of early virtual machines, aiming at exploring a solution to making most of hardware with tens or even hundreds of CPUs [1]. After the birth of Disco, many virtual machines of various architecture have emerged continuously, all more or less contributing to the prosperous market of virtual machines today. For a typical virtual machine, it should be a complete computer system with complete hardware system functions simulated by virtualization software in a completely isolated environment. All hardware operations which can be done in the physical computer also can be implemented in the virtual machine. However, not like those physical computers, the virtual machine monitor, also called the hypervisor, plays the role of a bridge between all virtual machines and the only host machine, allowing multiple virtual machines running simultaneously. The varied running environments of the hypervisor, which can be either inside the host operating system or directly on the hardware of the host, also determines the two different types of architecture: hosted architecture and bare-metal architecture. In this work, the former is represented by VMware Workstation and the latter is represented by Microsoft Hyper-V.

Despite the flourishing of virtual machines, more and more concerns about the system security of virtual machines have been raised since late 2000s, which justifiably gives an opportunity to the rapid development of an innovative technique termed container [1]. Compared with virtual machines, containers are not oriented to the virtualization of hardware for operating systems, instead, they completely focus on the virtualization of the deployment environment for applications. They allow users to download images and run them inside containers conveniently. Hence, they are far more lightweight in terms of size and faster in launching speed than any virtual machines. In essence, a container is a group of processes in a sandbox which is isolated and controlled by the host kernel. Currently, containers like Docker, have been widely used in deployment of applications since the concept came out.

This report aims to present and evaluate the performance of three popular personal virtualization solutions in Windows 10 host environment including VMware Workstation, Hyper-V and Docker. To perform tests, I use common tools (sysbench, fio, mbw) to conduct performance tests in different respects of hardware on these virtualization platforms on the same host machine. A comparison and evaluation of their performance in a preset virtual environment is in forms of tables and charts.

978-1-6654-0605-5/21/$31.00 ©2021 IEEE

2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC) | 978-1-6654-0605-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICFTIC54370.2021.9647226

II. Background

Since the characteristics of these virtualization approaches is apparently determined by their different architecture, it is important to have a deeper understanding of specific underlying principles. Among these three platforms, VMware Workstation and Hyper-V are typical platforms of the virtual machine with hosted and bare-metal hypervisor architecture respectively and Docker is the paradigm for virtualization at application level.

A. VMware Workstation

VMware Workstation is a powerful solution of personal virtualization, which combines the classic hosted architecture with other creative innovation. As a virtualization platform with the host architecture, VMware Workstation is required to be installed in the host operating system to create and manage virtual machines. In the host operating system, VMware Workstation application runs in the same way as any other programs that run in parallel. However, at underlying level, the virtualization environment is created by VMware Driver embedded in the host operating system, which provides complete hardware virtualization of the underlying hardware for those operating systems [2]. The VMware Workstation driver deploys the virtual machine manager that can run in a privileged mode so as to pass its requests to the host operating system by system calls and thus provide a bridge for VMware applications to dispose specific I/O requests [2]. Therefore, VMware Hypervisor has the authority to access the underlying hardware through the host operating system to serve the virtual machine instances it carries.

With this architecture, it is feasible to isolate virtual machine instances in assigned memory spaces and provide considerable performance since operations involving VMware intervention such as BT (binary translation) is only required when instructions on device I/O are involved. These instructions are managed by the hypervisor (VMM), which controls the CPU and the memory management unit (MMU), acting with the host OS alternatively.

B. Hyper-V

In 2008, Microsoft announced the releasing of a beta of Hyper-V, which is based on x86 hardware virtualization extensions [3]. It is a virtualization solution for Windows servers which applies the micro-kernel architecture to achieve better performance under the premise of security. When we analyze the structure of Hyper-V in a vertical point of view, we can find that the underlying Hypervisor of Hyper-V runs at the highest authority level, which Microsoft names ring-1. Correspondingly, the operating system kernel and drivers of the virtual machine run in ring-0, and the application runs in ring-3 [4]. With such a concise architecture, complicated BT (binary translation) procedure is dispensed with, and security of the host can be further improved. Hence, in the respect of architecture, Hyper-V has only three layers: hardware, hypervisor and virtual machine with no third-party drivers. Therefore, it may have the advantage of high execution efficiency, which is able to obtain a high utilization rate of hardware resources, enabling the virtual machine system performance that is close to physical system performance.

When we analyze the structure horizontally, it is apparent that different parts of Hyper-V isolates at the partition level. Each partition is a logical isolation unit which is managed by the hypervisor. The hypervisor must have at least one parent/root partition in order to create a space for VMMS (Virtual machine management system). The virtualized stack runs inside the parent partition and has direct access to hardware devices. Subsequently, once a virtual machine launches, the parent partition will create another child partition for the guest operating system by invoking the virtualized application programming interface (API). The child partition has no access to the hardware whereas it is able to access the virtual view of the processor and run in the virtual memory address area that is assigned to each guest partition.

C. Docker

Docker, as one of the pioneers in the history of container technology, has obtained numberless favor of the public for its great convenience since it came out in 2013. The conception of Docker can be sorted into three components: the image, the container and the repository. Docker images are read-only templates, from which Docker containers are generated. Images contain entire environments in which only those applications required by the user are installed. When an image is downloaded from the repository, such as Docker Hub, the user can subsequently create a Docker container, which provides the running space and resources of applications. The relationship between an image and a container is very similar to that between classes and instances in an object-oriented programming language. All the containers can be created, started, stopped or deleted, isolated from each other all the time. Actually, it is better to consider a container as a simple version of the Linux environment with applications running in it. Also, apart from downloading images, everyone is allowed to upload their personal images to the repository which the user has access to.

As for the architecture of Docker, it is typical client-server. The Docker client, which is controlled by the user, is responsible for the delivery of users' instructions such as "docker build" and "docker pull" to the Docker daemon, a crucial component in the Docker host which is able to deal with different operations of Docker containers and local images inside the Docker host [5]. The Docker daemon is always listening for any Docker requests. A daemon can also communicate with other daemons to manage Docker services. Both running on the same computer and connecting to a remote Docker host is acceptable. Furthermore, when the user needs to deploy new images, the host is also able to download directly from accessible registries which stores a great number of images uploaded by users.

The configuration of the host machine in these tests is as follows:

Table 1: the configuration of the host machine

Host OS	Windows 10
Host CPU	Intel i5-9300H 2.40GHz
Host RAM	Double SK Hynix DDR4 2400 8G

Host hard disk	SAMSUNG MZVLB512HAJQ-00000, SSD

The configuration of virtual machines is as follows:

Table 2: the configuration of the virtual machine

VM CPU	8 virtual cores (maximum)
VM RAM	2048MB
VM hard disk	20GB, as a single file

These performance tests are conducted on CPU performance, RAM performance and disk I/O performance of three virtualization solutions running in the same environment. As shown in Table 1 and Table 2, the number of CPU cores and the sizes of memory space and hard disk space are the same for both virtual machines, running on the same host with fixed configuration in order to control variables. It should be noticed that the running environment of Docker is WSL2 (Windows Subsystem for Linux 2) under Windows 10. Also, considering that Hyper-V is a product of Microsoft, such as Windows 10, there may be better support and optimization in Hyper-V. Therefore, in this experiment, apart from tests under Ubuntu 20.04 as the virtual machine operating system, I also evaluate their performance under Windows 10. Due to differences in systems, performance evaluation tools will also be different: in Linux, sysbench, fio and mbw are used, while for Windows 10, Fritz Chess Benchmark, Crystal Disk Mark 8 and AIDA64 are used.

III. Results and evaluations

A. CPU and RAM test under Ubuntu 20.04 and WSL2

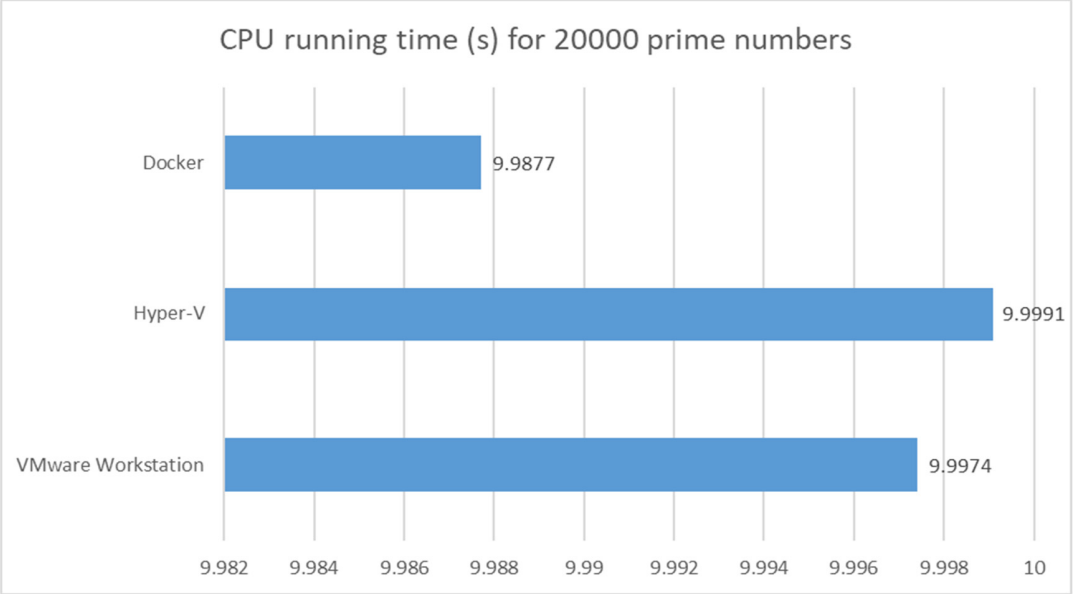

Figure 1: CPU running efficiency under Ubuntu 20.04

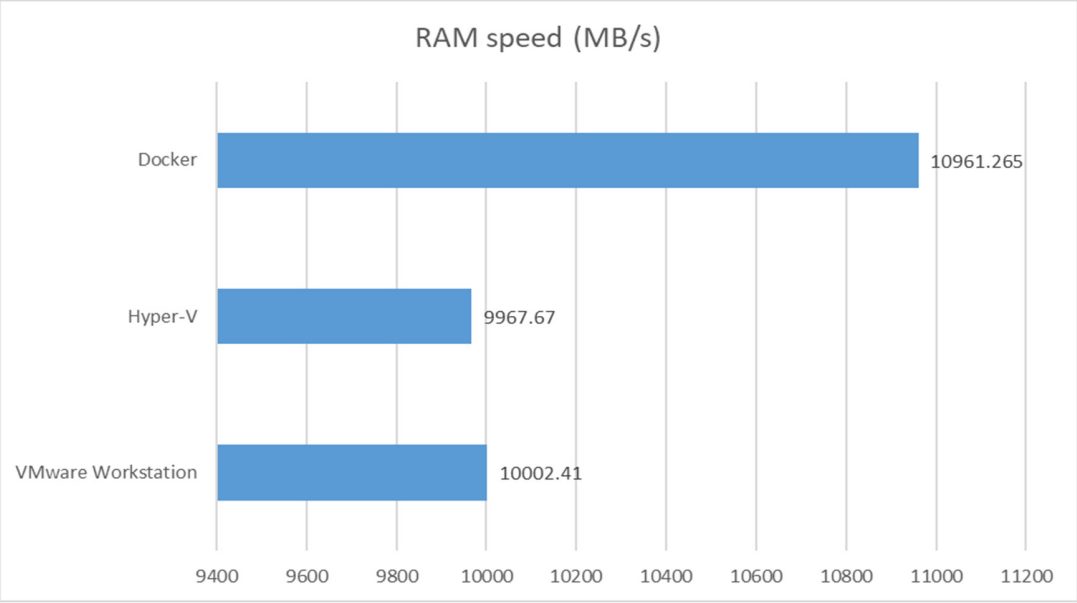

Figure 2: RAM efficiency under Ubuntu 20.04

Sysbench is a popular performance test tool under the Linux environment. The test rule of sysbench for CPU performance is simple: call the CPU to search for prime numbers in all integers until the number of prime numbers is equal to a preset value. In this test, I set the quantity of prime numbers to 20000.

According to Figure 1 and 2, it is obvious that the Docker container spend the least amount of time in finding these numbers and has the highest RAM speed, which is contributed to the tiny size and fewer abstraction layers compared to other two virtual machines. Because Docker does not require a hypervisor to virtualize any hardware resources, programs running on the Docker container use hardware resources directly from host machines.

However, compared with VMware workstation, Hyper-V as a virtualization platform with a bare-metal hypervisor, is for some reason outstripped by VMware Workstation in both tests.

B. Hard disk I/O test under Ubuntu 20.04 and WSL2

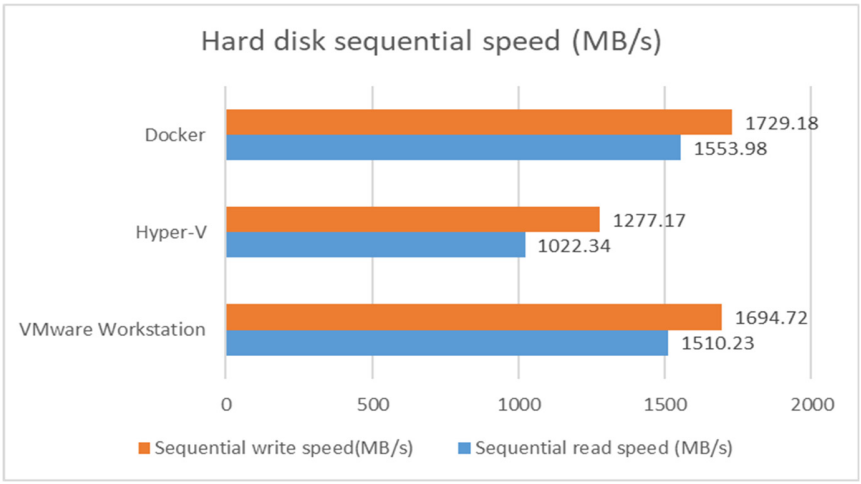

Figure 3: Hard disk sequential I/O efficiency under Ubuntu 20.04

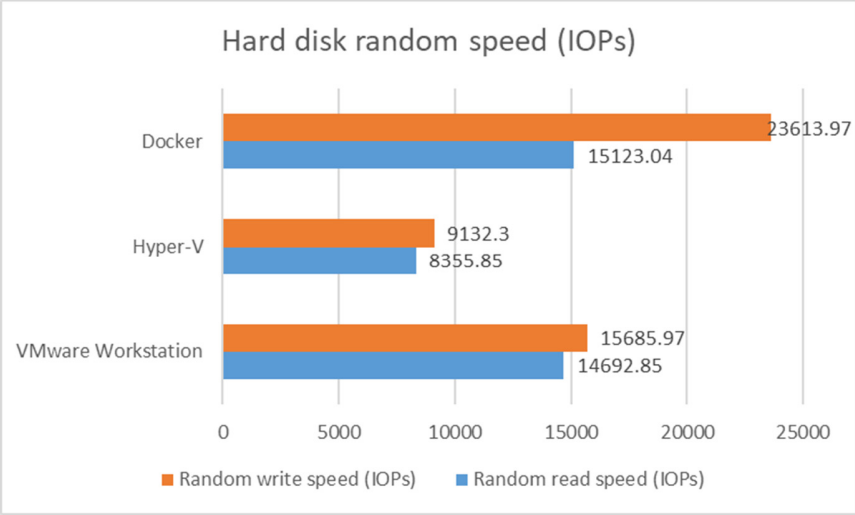

Figure 4: Hard disk random I/O efficiency under Ubuntu 20.04

For all three platforms, fio, a versatile I/O test tool, is applied to test both random and sequential hard disk performance. In these tests, the queue depth and the number of threads are set to 1. For the sequential part, the size of a file is set to 1 MB. For the random part, the size of a file is set to 4 KB.

According to Figure 3 and 4, the performance of Docker and VMware Workstation in sequential read and write is almost at the same level, and that of Hyper-V is apparently lower. Though quite close, the performance of Docker still beat that of VMware Workstation by a narrow margin. From another dimension, this result reflects the ingenious program optimization of VMware. However, in the test of random speed, although the performance of VMware Workstation is quite similar to that of Docker, Docker notably keeps a huge advantage: its random write speed is 1.5 times higher.

C. CPU and RAM test under Windows 10

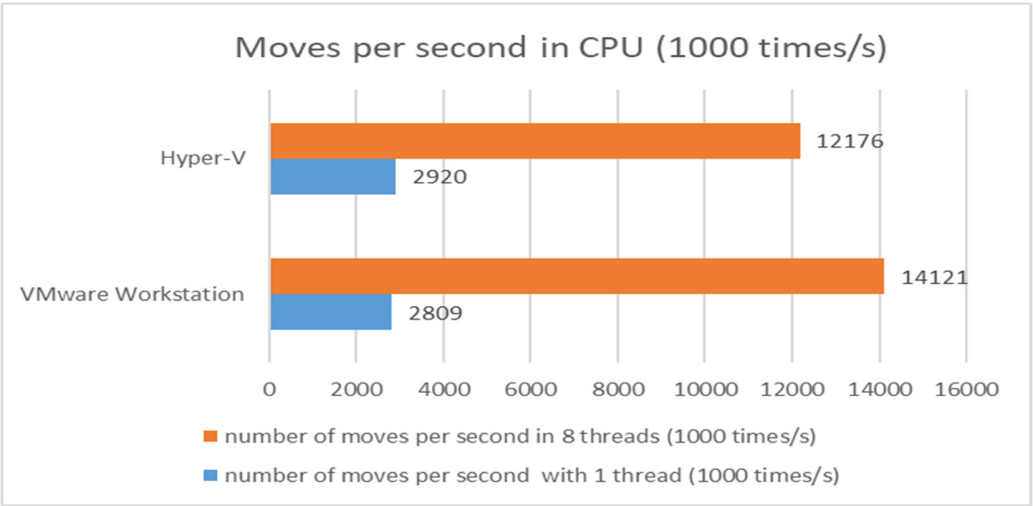

Figure 5: CPU running efficiency under Windows 10 environment

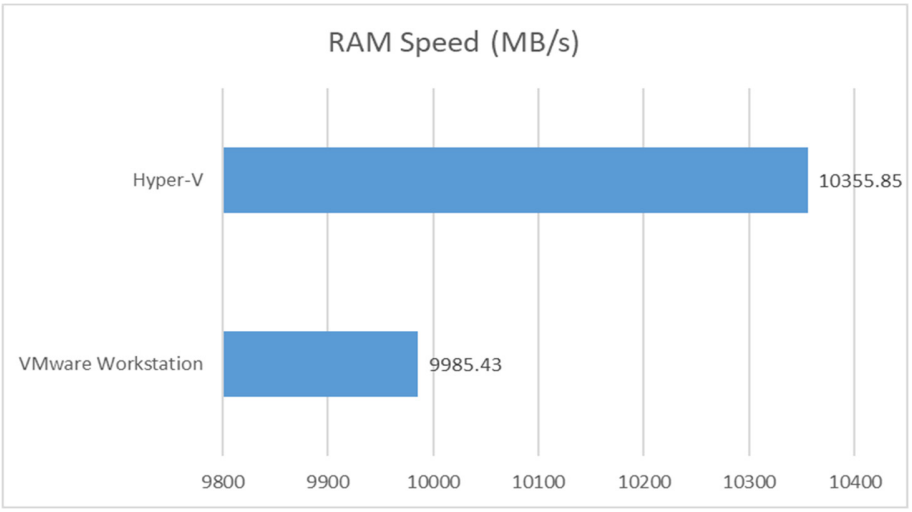

Figure 6: RAM efficiency under Windows 10 environment

Fritz Chess Benchmark is a unique CPU stress test tool for Windows that originated from a chess level testing software for humans. It tests the computation speed for the CPU by simulating the moves of a random chess game. After the test, it shows the final number of moves per second.

Before the test, I expected to see the better performance of Hyper-V under Windows 10. According to Figure 5 and 6, Hyper-V is more efficient than VMware Workstation in single-core performance and RAM speed, which is consistent with my expectations. In the field of multi-core computation speed, however, Hyper-V is surprisingly outstripped by VMware Workstation. This may be caused by the better multi-core optimization in the hypervisor of VMware, enabling the higher efficiency.

D. Hard disk I/O test under Windows 10

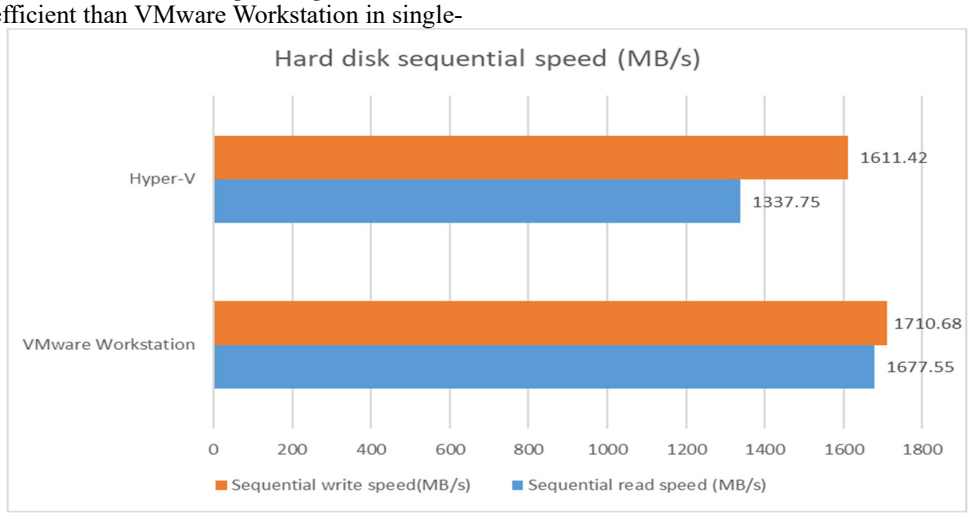

Figure 7: Hard disk sequential I/O efficiency under Windows 10

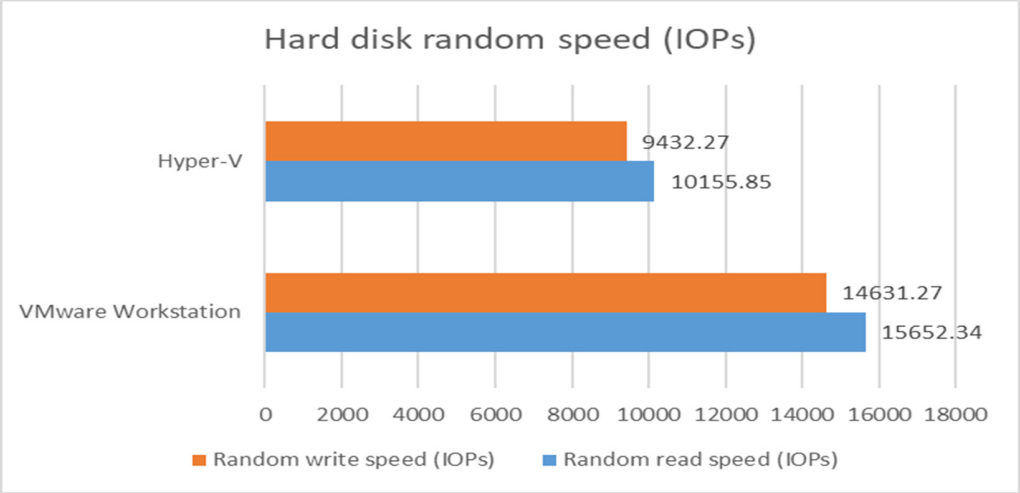

Figure 8: Hard disk random I/O efficiency under Windows 10

For the test of hard disk I/O under windows 10, Crystal Disk Mark, a simple I/O test tool, is applied to test both random and sequential hard disk performance on two platforms. In the test, the queue depth and the number of threads is set to 1. For the sequential part, I set the size of a file to 1MB. For the random part, I set the size of a file to 4KB.

According to Figure 7 and 8, although there is some improvement in both sequential tests and random tests, the performance of Hyper-V still does not meet my expectation to outstrip VMware Workstation: from every respect, Hyper-V performs worse.

IV. Conclusion

After the comparison among architecture of VMware Workstation, Hyper-V and Docker as well as their actual performance, their respective advantages and disadvantages have been demonstrated clearly. Among all three platforms, Docker has the best performance, which is contributed by its lightweight and flexible architecture. For VMware Workstation and Hyper-V under Ubuntu environment, the former performs better in all aspects, while the leading edge of it shrinks in terms of hard disk speed and RAM I/O speed under Windows 10 environment. This may be caused by the Hyper-V's better optimization for Windows, which is supported by Microsoft. However, except for the single-core performance in Fritz Chess Benchmark, Hyper-V does not show significantly more efficient virtual CPU computing performance than VMware Workstation in either the ubuntu environment or the windows 10 environment. This point is worthy of further exploration. Because as a virtual machine platform with bare metal architecture and optimization from Microsoft, Hyper-V should have higher operating efficiency at least under Windows 10.

Undoubtably, Hyper-V is a powerful free virtualization platform with updates for it provided by Microsoft. Nonetheless, as one of the leading products of current personal virtualization technology, VMware workstation could be a better personal virtualization solution when charges are not considered. It takes into account the rationality of both the user interface and high efficiency. In terms of CPU, memory and disk performance, it outstrips Hyper-V in most aspects in different operating systems. However, when users only need to simulate a specific application, it may be shackled by the underlying hardware simulation in the face of the lightweight and efficient container technology, not capable of giving play to its original advantages. Yet the advantages of the container like Docker also determine its disadvantages: it is not able to run like a computer. Instead, it is merely a sum of a simple running environment and some programs.

However, different virtualization technologies are never in competition with each other. In conclusion, only selecting the right virtualization technology according to specific needs can users achieve the highest efficiency at the least cost.

Reference

[1] Allison Randal (2020), The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers, in: ACM Computing Surveys. DOI: 10.1145/3365199.

[2] Edouard Bugnion, Scott Devene, Mendel Rosenblum, Jeremy Sugerman, Edward Y. Wang (2012), Bringing virtualization to the x86 architecture with the original VMware Workstation, in: ACM Transactions on Computer Systems. DOI: 10.1145/2382553.2382554.

[3] Jason A. Kappel, Anthony Velte, and Toby Velte (2009), Microsoft Virtualization with Hyper-V: Manage Your Datacenter with Hyper-V, Virtual PC, Virtual Server, and Application Virtualization, published by: McGraw Hill Professional.

[4] Borislav Djordjevic, Valentina Timcenko, Nenad Kraljevic, Nemanja Macek (2021), File System Performance Comparison in Full Hardware Virtualization with ESXi, KVM, Hyper-V and Xen Hypervisors. DOI: 10.4316/AECE.2021.01002.

[5] Marco De Benedictis, Antonio Lioy (2019), Integrity verification of Docker containers for a lightweight cloud environment, in: Future Generation Computer Systems. DOI: 10.1016/j.future.2019.02.026.