iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: X Month of publication: October 2021

DOI: https://doi.org/10.22214/ijraset.2021.38392

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Virtual Disks Performance Analysis Using Flexible I/O and Powerstat

Pritam Patange[1], Viraj Desai[2], Mahesh Shirole[3]
[1, 2, 3]Department of Computer Engineering, Veermata Jijabai Technological Institute, Mumbai, India

Abstract: Cloud computing has experienced significant growth in the recent years owing to the various advantages it provides such as 24/7 availability, quick provisioning of resources, easy scalability to name a few. Virtualization is the backbone of cloud computing. Virtual Machines (VMs) are created and executed by a software called Virtual Machine Monitor (VMM) or the hypervisor. It separates compute environments from the actual physical infrastructure. A disk image file representing a single virtual machine is created on the hypervisor's file system. In this paper, we analysed the runtime performance of multiple different disk image file formats. The analysis comprises of four different parameters of performance namely- bandwidth, latency, input-output operations performed per second (IOPS) and power consumption. The impact of the hypervisor's block and file sizes is also analysed for the different file formats. The paper aims to act as a reference for the reader in choosing the most appropriate disk file image format for their use case based on the performance comparisons made between different disk image file formats on two different hypervisors – KVM and VirtualBox.
Keywords: Virtualization, Virtual disk formats, Cloud computing, fio, KVM, virt-manager, powerstat, VirtualBox.

I. INTRODUCTION

Due to the pervasive use of the cloud in the previous decade, the performance of cloud-based systems has been in the limelight for quite some time. Virtualization is responsible for the paradigm shift from physical infrastructure to logical infrastructure. System designers now take into consideration the logical divisions of the physical infrastructure to a great extent in order to optimize the performance and resource usage. Virtualization creates a virtual version of a device or resource, such as a server, storage device or even an operating system where the framework divides the host operating system into more than one execution environments [1]. The most basic example of virtualization is the partitioning of a physical hard drive into multiple logical drives which can perform independently. In the simplest terms, virtualization is the technique that divides a physical computer into several isolated and independent machines known as Virtual Machines or VMs. The virtual file image created is stored on the hypervisor's local file system and gets loaded and executed form there itself. Different virtual disk image file formats are available for the different VMMs. Disk image file is a replica of disk drive assigned to one or more virtual machines; it works as a local hard disk for the VM or the guest operating system. The user can create a virtual disk image up to the size of the local file system. The number of virtual CPUs in a virtual machine cannot exceed the number of logical cores on the host [2]. As shown in Fig. 1, there are two types of hypervisors: Type 1 hypervisors which also called "bare metal", run directly on the system hardware; and Type 2 hypervisors which run on a host operating system that provides I/O device support, memory management and other virtualization services [3]. The main advantage of Type 2 hypervisors is that they do not require any special hardware for execution. Microsoft Hyper-V and Citrix/Xen Server are Type 1 hypervisors, while Oracle VM and KVM are type 2 hypervisors.

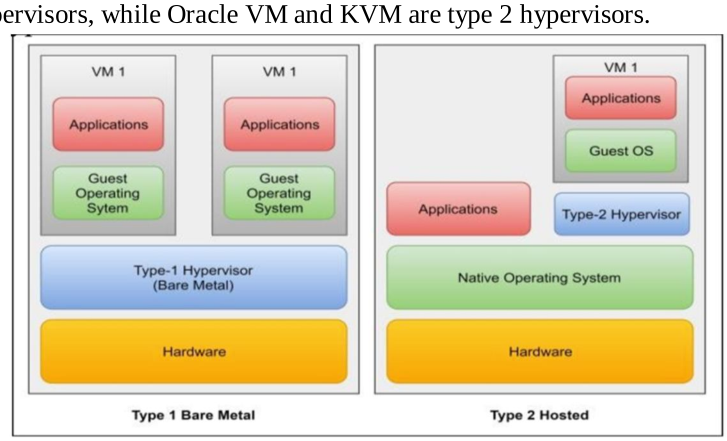

Fig. 1. Types of hypervisors

In this paper, we have performed experiments on different virtual disk file system environment which are created on two different hypervisors Oracle VM VirtualBox and KVM. Virtual disks QED (QEMU enhanced disk), Copy-on-Write scheme QCOW2 from QEMU, VMWARE's VMDK, RAW, Virtual box's VDI, Microsoft's VHD and VPC, and parallel's HDD are compared based on bandwidth, latency, IOPS and power consumption in watts. Various experiments were conducted to study the performance of each disk image file to derive performance comparisons against each other. We also investigate the power consumption model on different virtual disk file formats. The power equation can be written as equation (1)

$$P_{total} = P_{CPU} + P_{mem} + P_{io} + P_{\alpha} \quad (1)$$

Where P_{total} is the estimated total power consumption of system, P_{CPU} is the CPU or processing power consumption which has been defined and successfully modeled in the work of Anton Beloglazov et al. [4]. P_{mem} is the memory power consumption, P_{io} is the I/O power consumption of the disk and P_{α} is the power which is consumed by an idle background system which is not used in this paper since it is kept constant during the experiments. Only the P_{io} is taken into consideration for the power consumption parameter. The remaining paper has been divided into the following sections. Related work is mentioned in Section 2. The disk performance evaluation strategy is described in Section 3. The results of the study with a detailed analysis and charts are present in Section 4. Lastly, the conclusion of the paper and future scope are provided in Section 5.

II. RELATED WORK

Virtualization is the generic term used to represent a virtual entity as a physical entity [5]. This could also be sharing a single physical device into multiple virtual devices. A virtual computer created by virtualization is a logical disk computer that is used to perform all operations performed by an actual computer. Specialized software, called a hypervisor or VMM, emulates physical computer's processors, memory (RAM), hard disk (ROM) and other hardware resources completely which enables the virtual machines to share the resources among themselves.

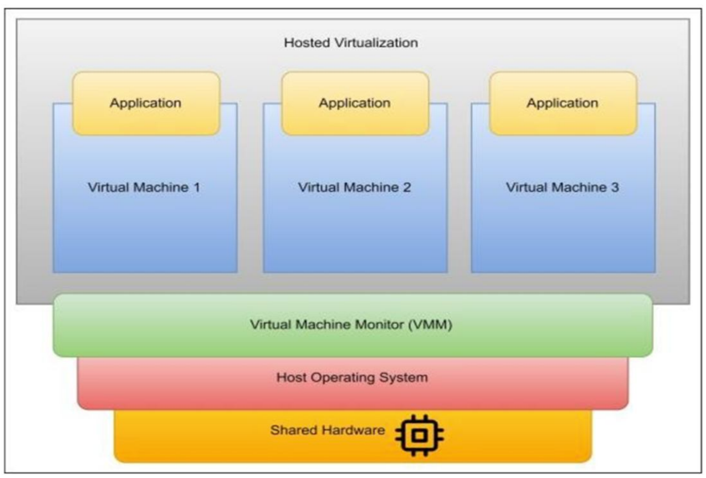

Fig. 2. Virtual Machine Environment

Fig. 2 shows a computer or a host machine on which the hypervisor is running one or more virtual machines called guest machines. KVM hypervisor relies on hardware virtualization technology to achieve better performance [6]. It was found that Xen3.1 holds a better performance than KVM [7]. KVM is a hardware- assisted virtualization tool developed by Qumranet Inc. [8] and VMware tool which uses full virtualization and is based on the x86 platform [9].

A. Flexible I/O

fio is short for Flexible I/O. The goal of fio was to save the hassle of writing special test case programs for testing particular workload and to replicate a bug or test the performance of the input on the testbed. fio is used to avoid writing test cases repeatedly for simulating IO workloads [10]. fio is a versatile IO workload generator. Owing to its flexibility, fio is one of the tools which is used to benchmark SSD or HDD I/O. fio takes three inputs, block size, file size and IO pattern and gives three outputs latency, bandwidth and IOPS. Sequential read and write tests, as well as random read and write tests can be executed along the option to use various block sizes from small 4KB to huge sizes of 1MB and more.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

B. Powerstat

One of the programs through which we can capture and measure the power consumption of any device such as a smartphone or a computer using a battery as the power source is Powerstat. Unlike vmstat, the statistics of power consumption are also calculated. The standard deviation, average and min/max of the gathered data are calculated by Powerstat, once it finishes it's execution. It is a tool to measure laptop power consumption. It basically monitors the system for 10 seconds intervals which are spread over a total of 480 seconds resulting in the collection of 48 different samples. While executing, it shows information such as time, user, nice, watts etc. which is fetched from the Kernel's output [11]. Powerstat takes the performance load generated by fio as input and gives watts consumed per sec as output.

III.METHODOLOGY

Fig. 3 shows the experiment test bed we set up for the study. We configured two hypervisors KVM and VirtualBox to use the various virtual disk image file formats like VDI, VHDK, QCOW to name a few. These virtual disk configurations and application workloads were executed on this testbed.

A. Experiment Test Bed for Performance Study

We have set up experiment testbed to be one machine with the following configuration: Intel Core i3-4005U, Clock Speed- 1.70 GHz and RAM is 4GB DDR3. Six Virtual machines will be created in this machine with 200GB volume and 2048MB RAM for each on two hypervisors KVM and VirtualBox separately.

B. Proposed Architecture

We have used I/O micro-benchmark workload to characterize the VM runtime performance and powerstat tool formats under the same condition.

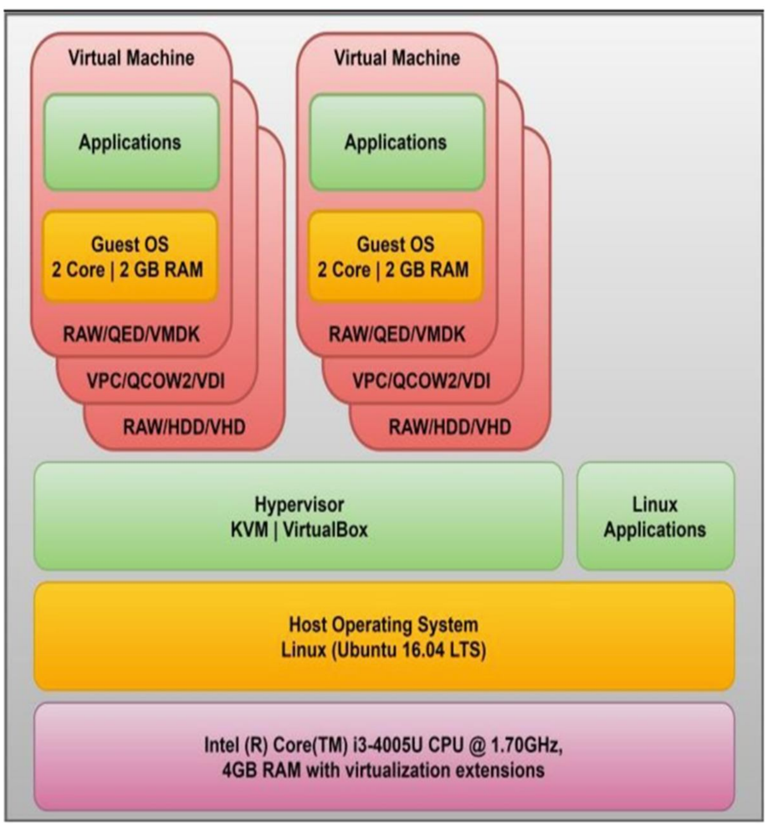

Fig. 3. Test Bed for Performance Study

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

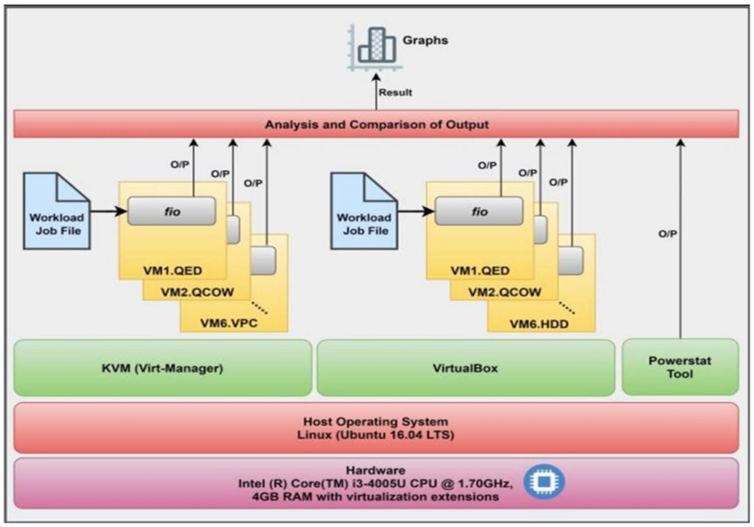

Fig. 4. Block Diagram for Proposed Study

The virt-manager tool allows uses a graphical interface to interact with KVM hypervisor [12]. As shown in Fig. 4, on hardware using two hypervisors KVM and VirtualBox multiple virtual disks of different file format will be installed separately due to the limitation of CPU cores on the host machine and workload job file, will be given to fio benchmark on each virtual machine as input. The result of fio benchmark will be gathered. Powerstat takes performance load generated by fio and Linux applications of the guest OS on each VM as input and gives watts consumed per sec as output. The result of the powerstat tool will be gathered. The analyzed result of fio and powerstat act as an reference to virtual disk end users for creating and using them efficiently.

In this experimental study, fio benchmark tool takes seven parameters as input using workload job file which is saved in text format. In the workload job file, four out of the seven parameters are constant, while three parameters assume different values and are variable. In Fig. 6, the parameters which are underlined with red are variable, while rest of them have constant values. On dividing the job file from top to bottom, the parameters as shown in Fig. 5 are identified.

```
;START
[seq-write]
ioengine=libaio
iodepth=4
rw=write
bs=8k
direct=0
size=1024m
numjobs=1
;END
```

Fig. 5. Sample fio job file

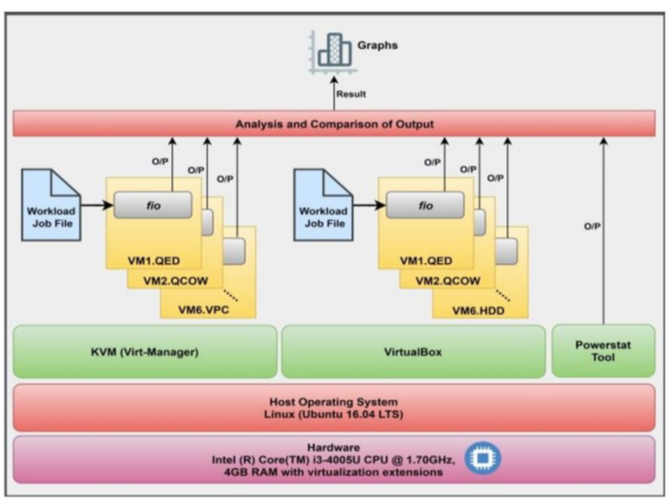

Fig. 6. Process bench marking fio with configuring parameters

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

I/O engine (ioengine): a way of IO issue
I/O depth (iodepth): the amount of IO units that will simultaneously hit a file(s) with requests [13]
I/O type (RW): Defines the I/O pattern issued to the file(s) Block size (bs): size of IO operation chunk
Direct I/O (direct): an IO access where you bypass the cache. I/O size (size): I/O file size.
Target file/device (numjobs): number of files, we are spreading over workload.
In this experimental study, we have tested the application runtime performance on each hypervisor based on the following parameters: Block Sizes - 4KB to 128KB, File Sizes - 64MB to 2048MB, IO Patterns used - Sequential Read, Sequential Write, Random Read and Random Write.

IV. EXPERIMENTAL RESULTS

The results are analyzed based on latency measured in milliseconds (ms), bandwidth measured in KiloBytes per second (KB/s), IOPS and power measured in watts. Latency, bandwidth and IOPS are collected from fio benchmark output, whereas power consumptions in watts are collected from the powerstat tool. It is observed that the results vary in relation to the changes in file size, block size and I/O pattern in the job file. So, we have performed several tests to capture the variations in the results for each virtual disk image file format. In all experiments conducted the inputs included different combinations of block sizes from 4K to 128K and file sizes from 64MB to 2048MB.

A. Performance Results

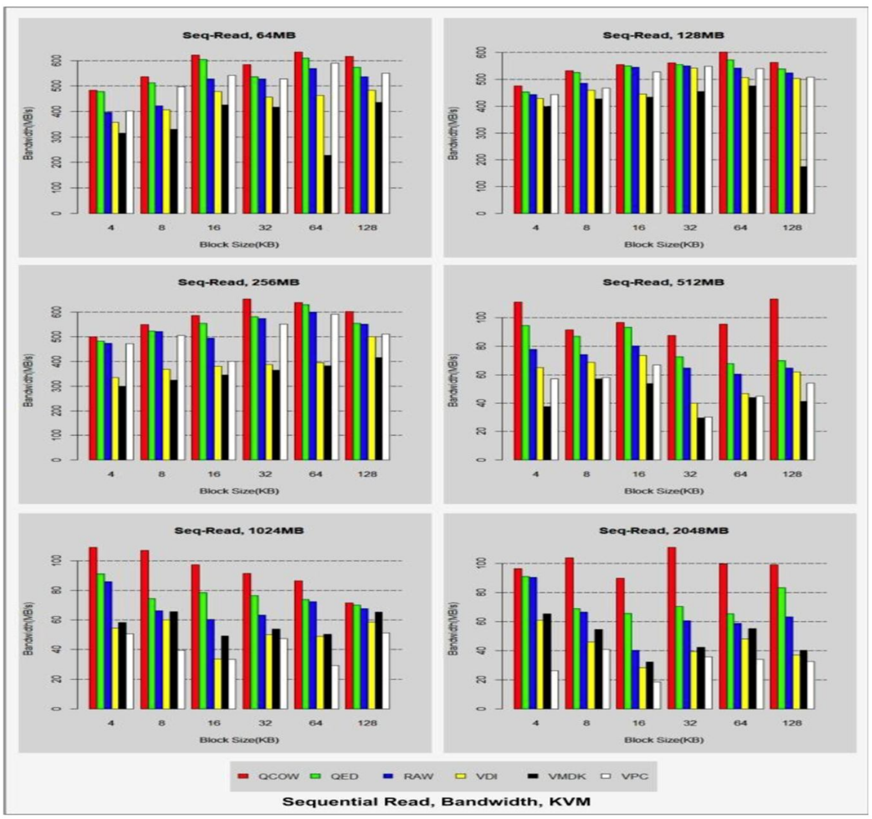

Fig. 7. Bandwidth parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 7 shows the bandwidths for different disk image file formats for sequential read input on KVM hypervisor. QCOW and QED formats perform comparatively well for all file sizes and for all block sizes. For smaller file sizes, all virtual disk images perform well. As file size increases, most of the virtual disk images see a decrease in performance. As file size increases the performance of VMDK virtual disk increases whereas the performance of VPC decreases. The overall preferable sequence of disk image formats for a sequential read based on bandwidth parameter on KVM hypervisor is QCOW, QED, VPC, RAW, VDI and VMDK for file sizes smaller than 512MB and for larger file sizes the preference order is QCOW, QED, RAW, VMDK, VDI and VPC.

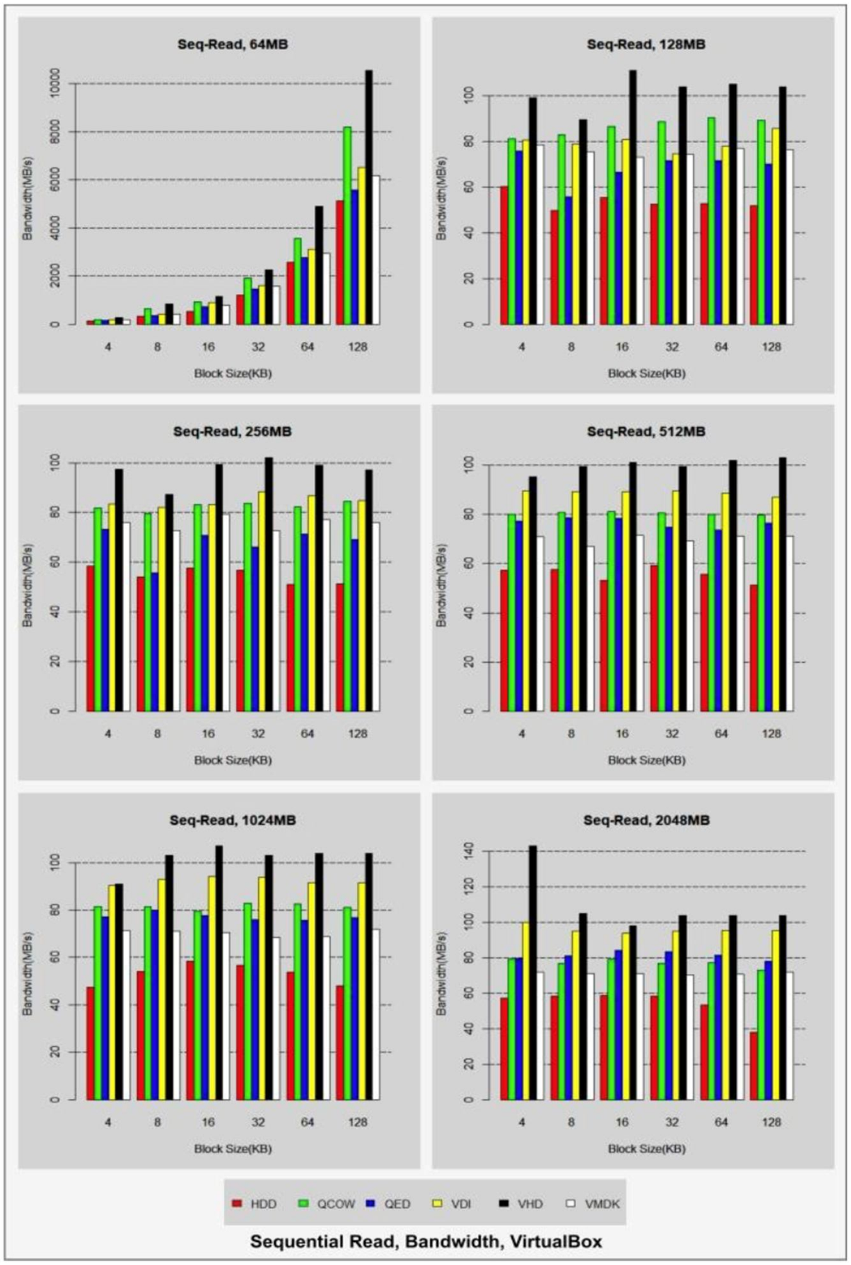

Fig. 8. Bandwidth parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB VirtualBox hypervisor.

Fig. 8 shows the bandwidths for different disk image file formats for sequential read input on VirtualBox hypervisor. VHD and QCOW perform relatively well for all file and block sized, whereas QED and HDD perform worst. As the file size increases, QCOW's performance relatively decreases, whereas the performance of VDI increases. The overall preferable sequence of virtual disk image file formats for a sequential read based on bandwidth parameter on VirtualBox hypervisor is VHD, QCOW, VDI, VMDK, QED and HDD for file sizes below 256MB and for larger file sizes the preference order is VHD, VDI, QCOW, QED, VMDK and HDD.

Based on above results for sequential read pattern, we can infer that disk image file formats which perform well have bandwidth of 80 MB/s and above, those with average performance have bandwidth ranging from 60 to 79 MB/s, whereas disk image file formats which perform poorly have bandwidth of 60 MB/s and less. The overall preferable virtual disk image file format for a sequential read based on bandwidth is QCOW among both hypervisors.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

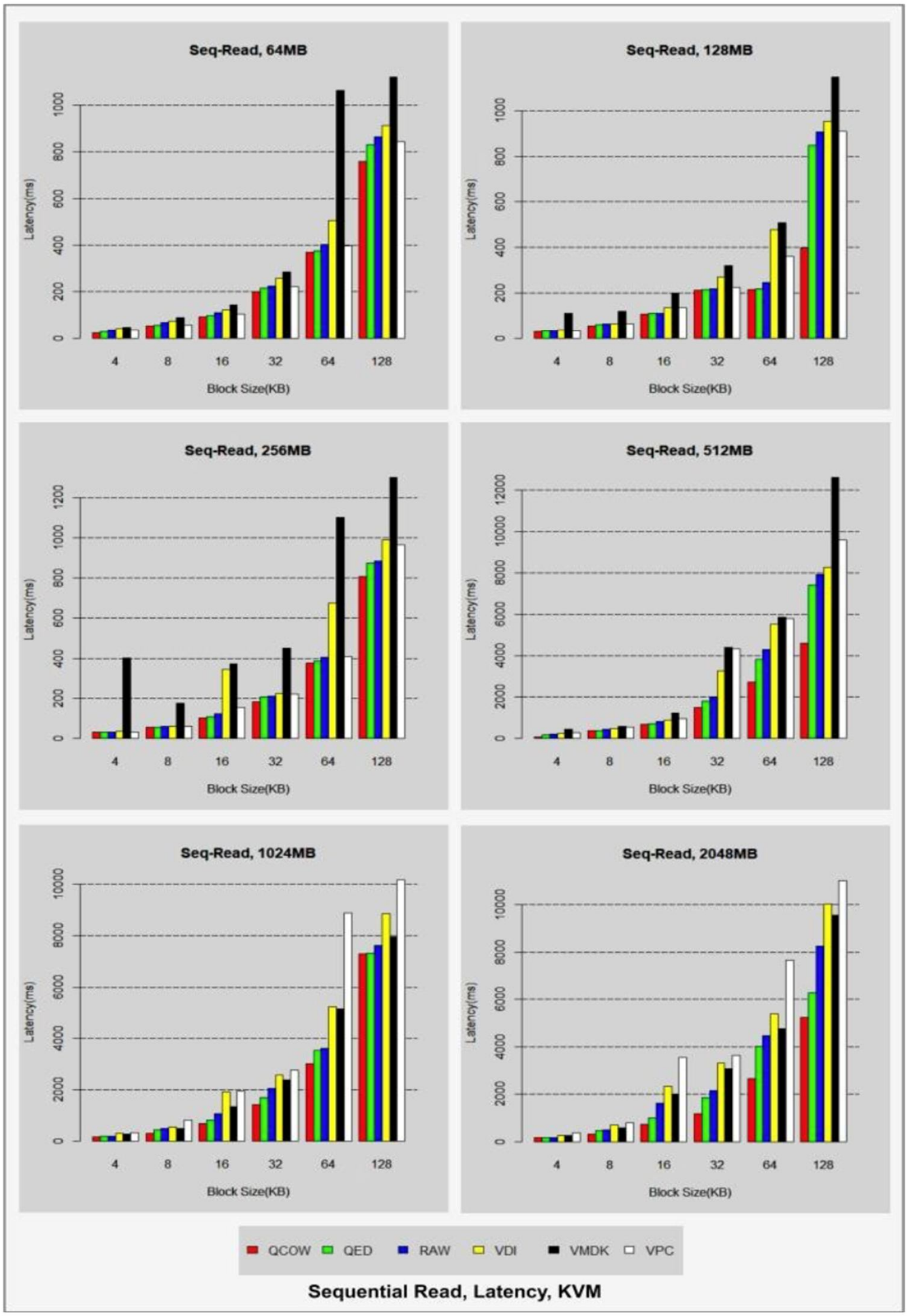

Fig. 9. Latency parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 9 shows the latencies for different disk image file formats for sequential read input on KVM hypervisor. QCOW and QED virtual disk perform well for all file sizes and on all block size whereas VMDK performed the worst. For small file sizes all virtual disk images perform well, but as file size increases, all formats observe a performance decrease. The performance of all disk image file formats is inversely proportional to the input block size. As file size increases the performance of VMDK virtual disk increases, whereas the performance of VPC decreased. The overall preferable sequence of virtual file formats for a sequential read operation based on latency, on KVM hypervisor is QCOW, QED, VPC, RAW, VDI and VMDK

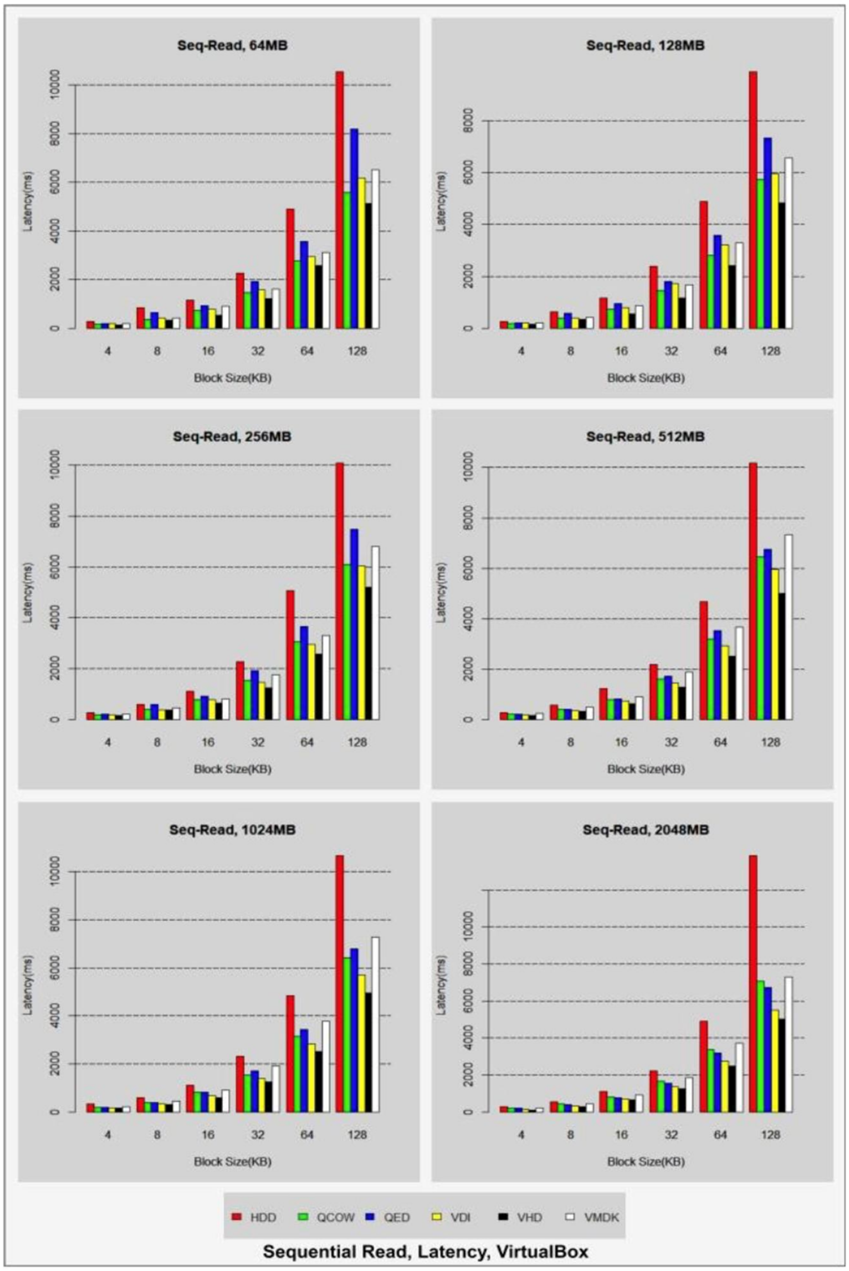

Fig. 10. Latency parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Fig. 10 shows the latencies for different disk image file formats for sequential read input on VirtualBox hypervisor. VHD and QCOW perform well for all file size and block size, whereas QED and HDD perform worst. As file size increases, QCOW virtual disk image's performance decreases, whereas the performance of VDI virtual disk increases. The overall preferable sequence of virtual file formats for a sequential read operation based on latency, on VirtualBox hypervisor is VHD, QCOW, VDI, VMDK, QED and HDD.

Based on above results for a sequential read operation, we can infer that disk image file formats which perform well have latency of 2000 ms and less, those with average performance have latency between 2000 and 5999 ms, whereas disk image file formats which perform poorly have latency of 6000 ms and above. The overall preferable virtual disk image file format for a sequential read based on latency is QCOW among both hypervisors.

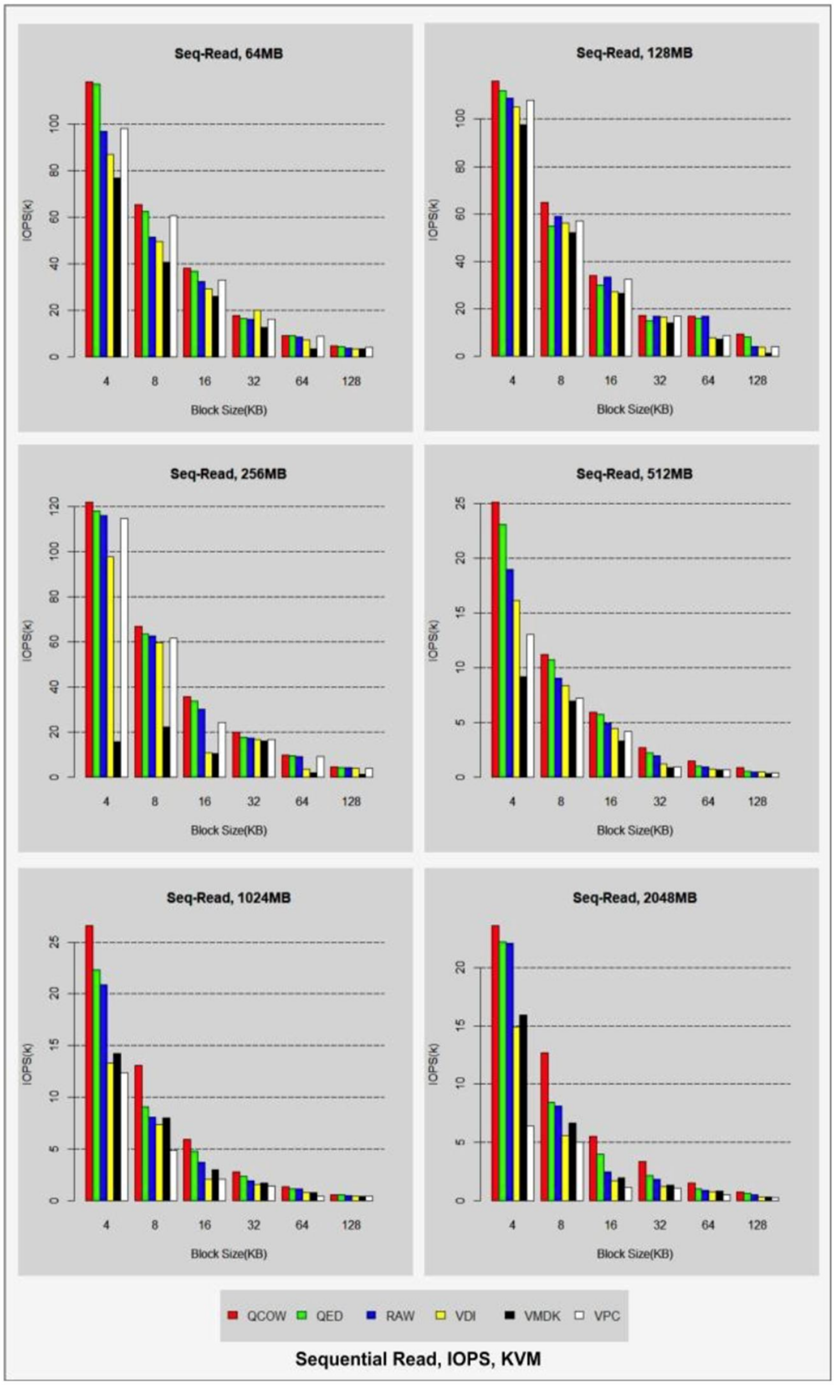

Fig. 11. IOPS parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 11 shows the IOPS for different disk image file formats for sequential read input on KVM hypervisor. QCOW and QED perform well for all file sizes and on all block sizes. For small file sizes, all disk image file formats perform well. As file size increases the performance of all image formats decrease. The performance also decreases for all image for- mats as the block size increases. With increasing file sizes, the performance of VMDK increases, whereas the performance of VPC decreases. The overall preferable sequence of virtual disk image formats for a sequential read based on IOPS, on KVM hypervisor is QCOW, QED, RAW, VPC, VDI and VMDK for file sizes smaller than 256MD and for larger file sizes the preferable sequence is QCOW, QED, RAW, VDMK, VDI and VPC.

Fig. 12 shows the IOPS for different disk image file formats for sequential read input on VirtualBox hypervisor. VHD and VDI have the best performance for all file size and block sizes, whereas QED and HDD perform worst. As the file size increases, the performance of QCOW virtual disk image decreases, whereas the performance of VDI virtual disk increases. The overall preferable sequence of virtual hard disks for a sequential read, IOPS on VirtualBox hypervisor is VHD, QCOW, VDI, VMDK, QED and HDD.

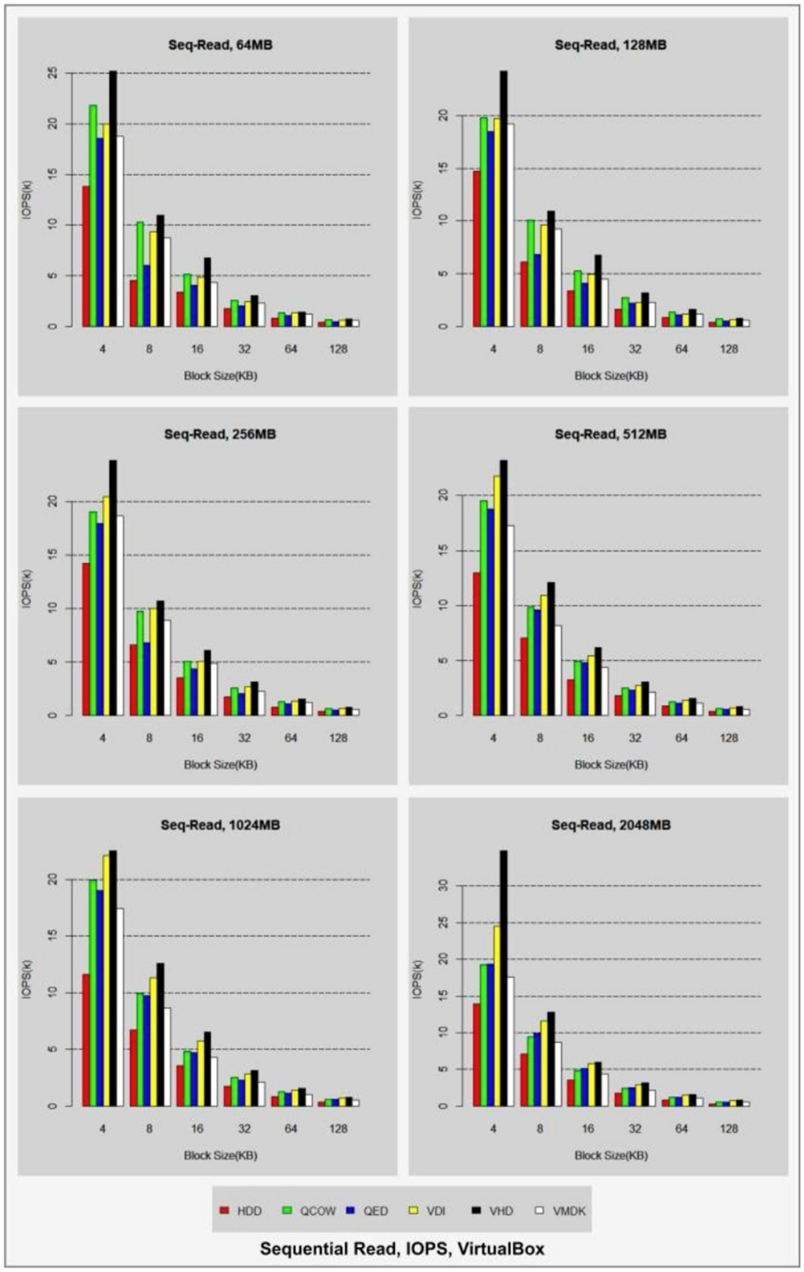

Fig. 12. IOPS parameter analysis for sequential read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Overall KVM performs well than VirtualBox hypervisor for sequential read pattern. Based on above results for sequential read pattern, we can infer that disk image file formats which perform well have IOPS of 20k and above, those with average performance have IOPS between 10k and 20k, whereas disk image file formats which perform poorly have IOPS of 10k and less. The overall preferable virtual disk image format for a sequential read based on IOPS among both hypervisor is QCOW. VHD performs best and HDD performs worst on VirtualBox hypervisor while QCOW performs best and VPC performs worst on KVM hypervisor.

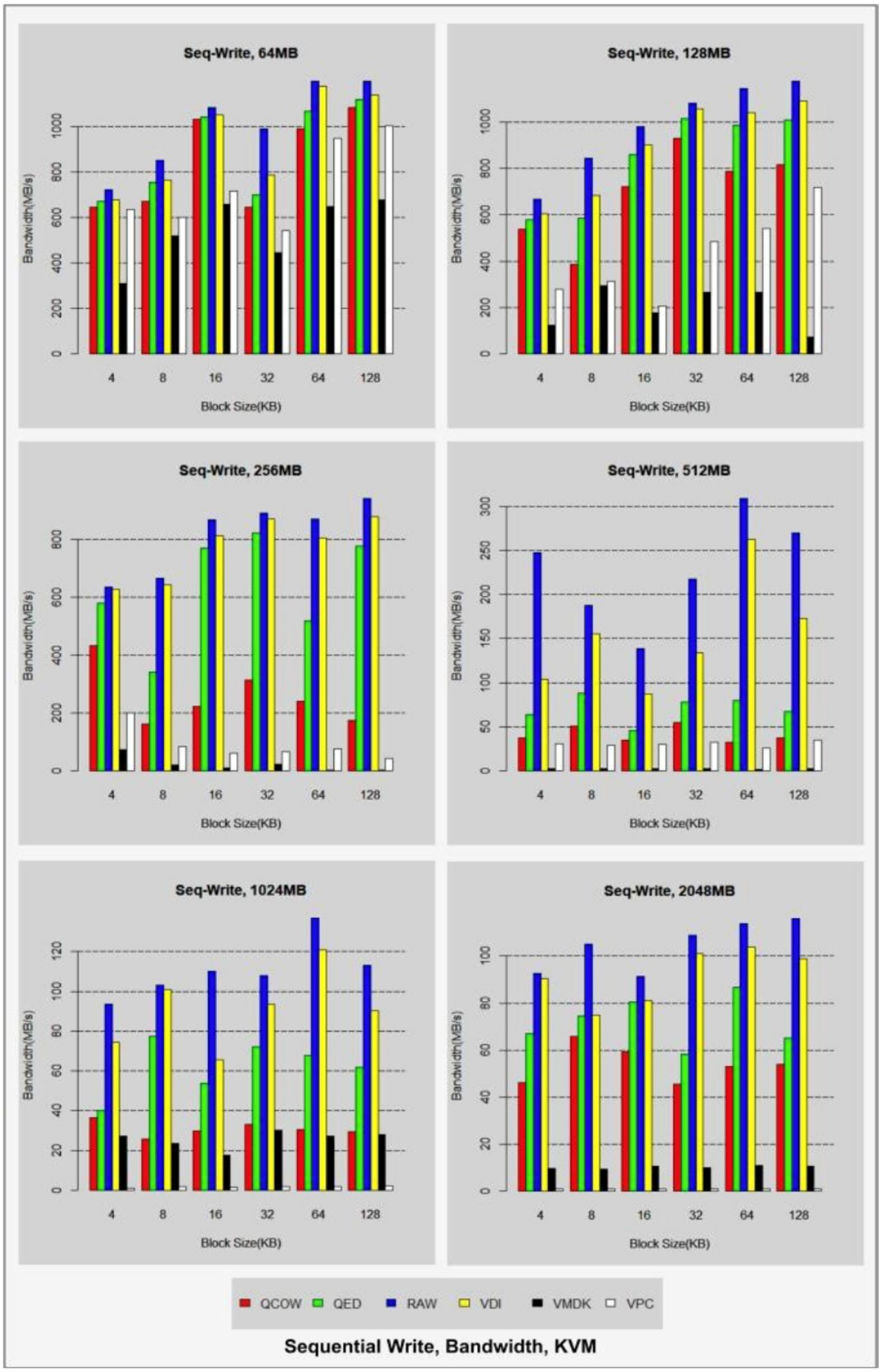

Fig. 13. Bandwidth parameter analysis for sequential write on block sizes 4KB-128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 13 shows the bandwidths for different disk image file formats for sequential write input on KVM hypervisor. Both RAW and VDI perform relatively well for all file sizes and on all block sizes. For small file sizes, all virtual disk image formats perform well, but the performance increases slightly as the block size increases. With increase in file size, the performance of all virtual disk images decrease. As the file size increases, the performance of VMDK relatively increases, whereas the performance of VPC decreased drastically. The overall preferable sequence of virtual image file formats for a sequential write based on bandwidth, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

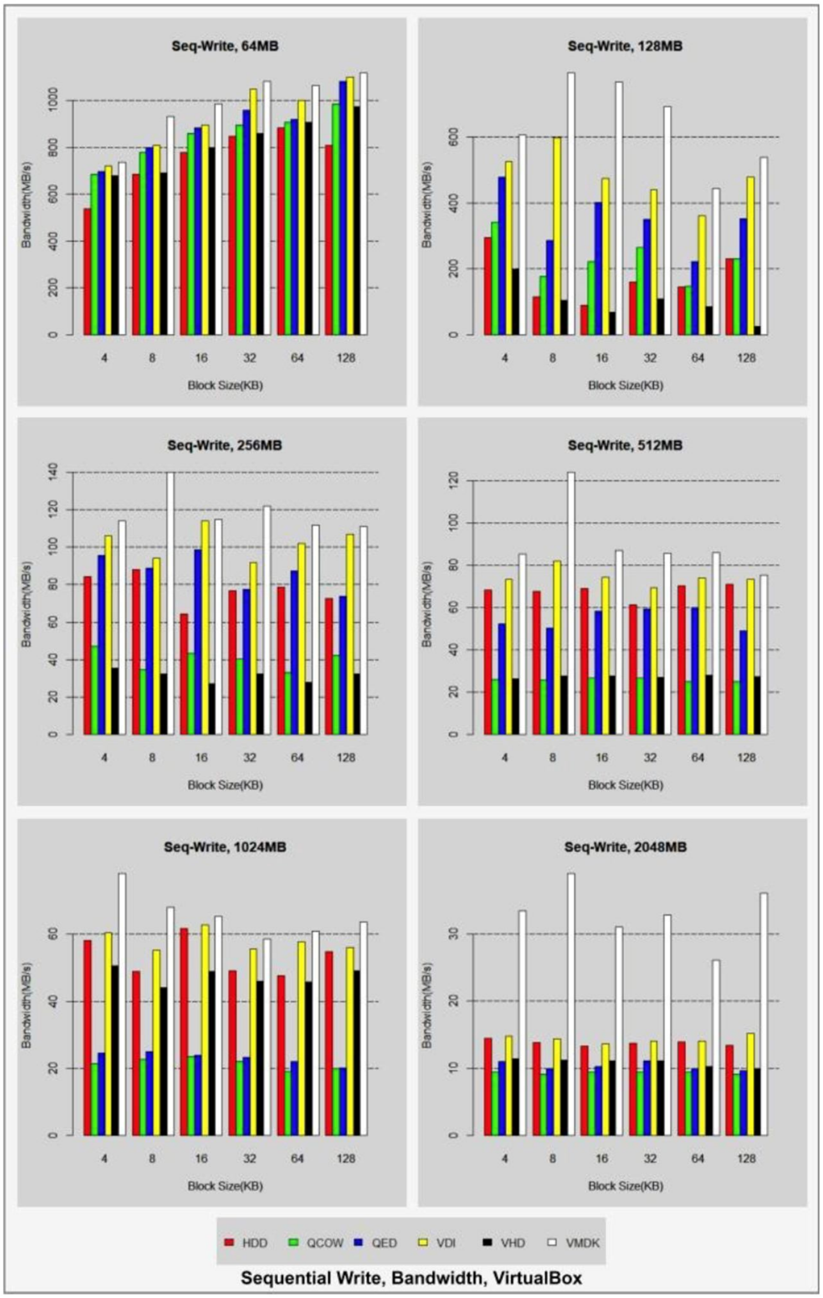

Fig. 14. Bandwidth parameter analysis for sequential write on block sizes 4KB-128KB for different file sizes 64MB – 2048MB VirtualBox hypervisor.

Fig. 14 shows the bandwidths for different disk image file formats for sequential write input on VirtualBox hypervisor. VMDK and VDI perform well for all file sizes and all block sizes, whereas VHD and HDD perform worst for small file sizes and for larger file sizes QCOW and QED perform the worst. As the file size increases, of QCOW's performance decreases, whereas VHD virtual disk's performance increases. The overall preferable sequence of virtual disk image file formats for a sequential write based on bandwidth, on VirtualBox hypervisor is VMDK, VDI, HDD, VHD, QED and QCOW.

Based on above results for sequential write pattern, we can infer that disk image file formats which perform well have bandwidth of 80 MB/s and above, those with average performance have bandwidth ranging from 60 to 79 MB/s, whereas disk image file formats which perform poorly have bandwidth of 60 MB/s and less. The overall preferable virtual hard disks for a sequential write based on bandwidth on both hypervisor is VDI. VMDK performs best on VirtualBox hypervisor, while RAW performs best on KVM hypervisor.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

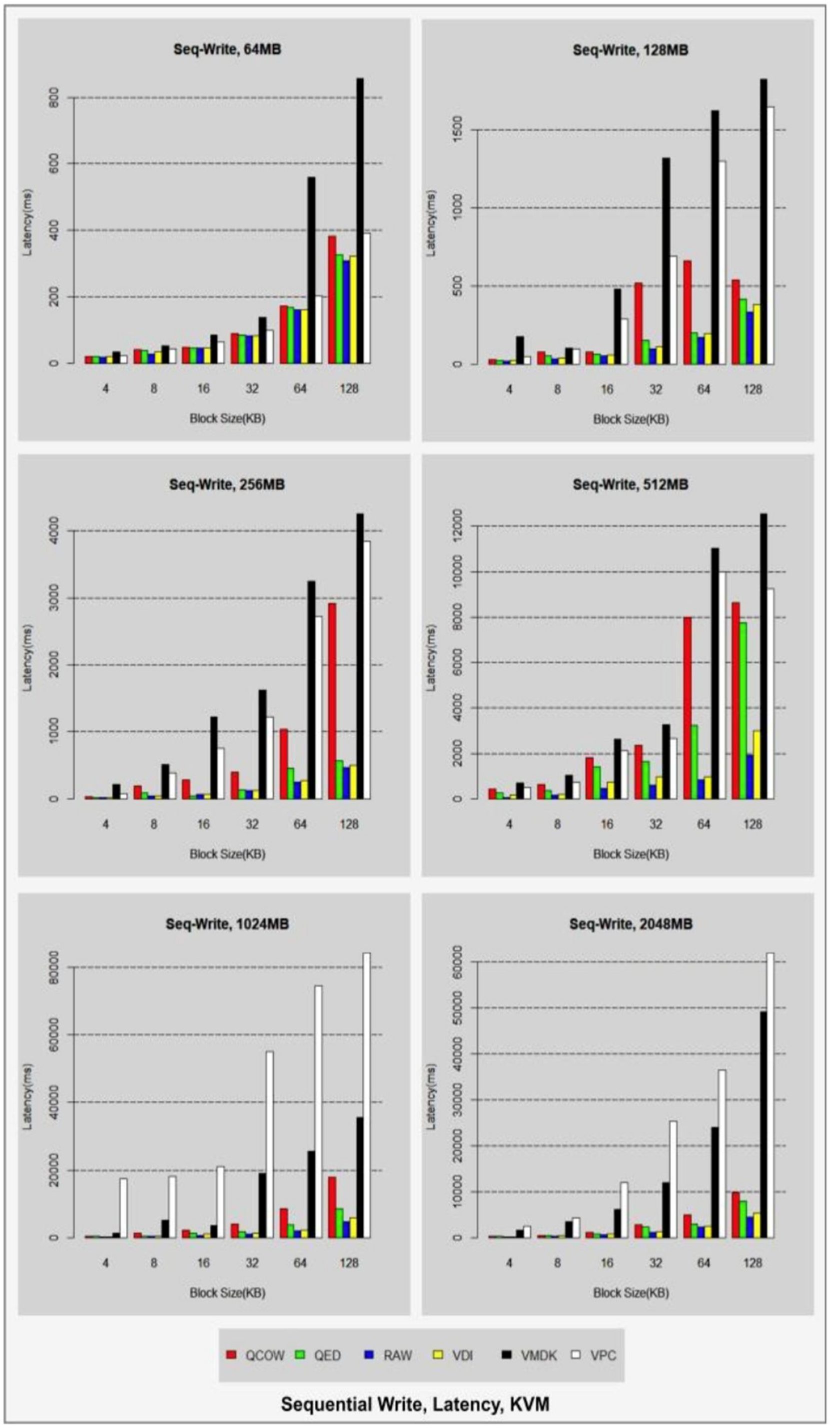

Fig. 15. Latency parameter analysis for sequential write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 15 shows the latencies for different disk image file formats for sequential write input on KVM hypervisor. RAW and VDI virtual disk image formats perform well for all file sizes and on all block sizes. For small file sizes, all virtual disk images perform well, but performance decreases i.e., latency increases as the block size increases. With an increase in file size, all virtual disk images observe an decrease in performance. With increase in file size, VMDK virtual disk's performance increases in comparison to other file formats, whereas the performance of VPC decreases in comparison to others. The overall preferable sequence for virtual disk image file formats for a sequential write based on latency, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Fig. 16 shows the latencies for different disk image file formats for sequential write input on VirtualBox hypervisor. VMDK and VDI perform well for all file sizes and for all block sizes, whereas VHD and HDD perform worst for small file sizes and for larger file sizes QCOW and QED perform the worst. With an increase in the file size, QCOW virtual disk image's performance decreases significantly. The overall preferable sequence of virtual disk image file format for a sequential write based on latency, on VirtualBox hypervisor is VMDK, VDI, HDD, VHD, QED and QCOW.

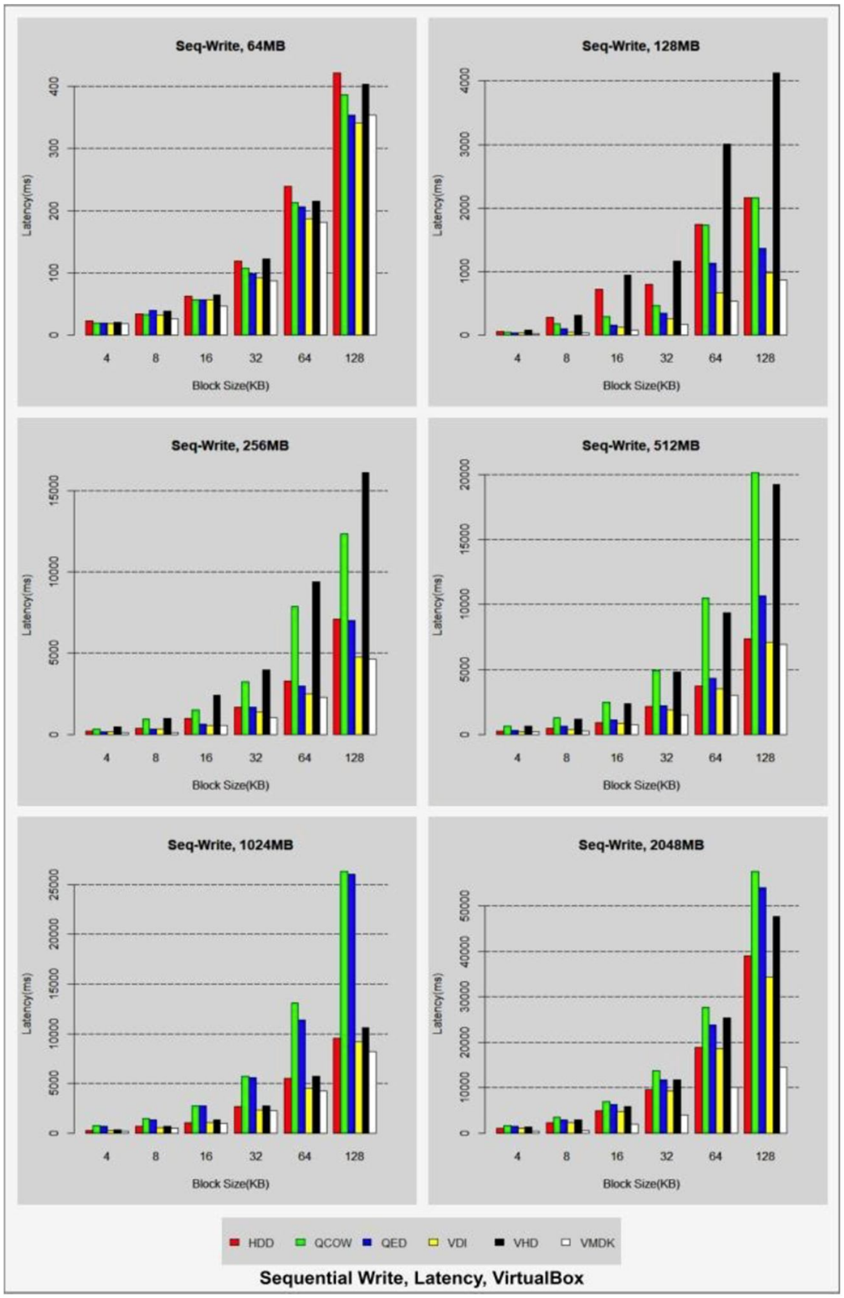

Fig. 16. Latency parameter analysis for sequential write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for sequential write pattern, we can infer that disk image file formats which perform well have latency of 2000 ms and less, those with average performance have latency between 2000 and 5999 ms, whereas disk image file formats which perform poorly have latency of 6000 ms and above. The overall preferable virtual hard disks for a sequential write based on latency on both hypervisors is VDI. Among both hypervisors, VMDK performs best on VirtualBox hypervisor and RAW performs the best on KVM hypervisor.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

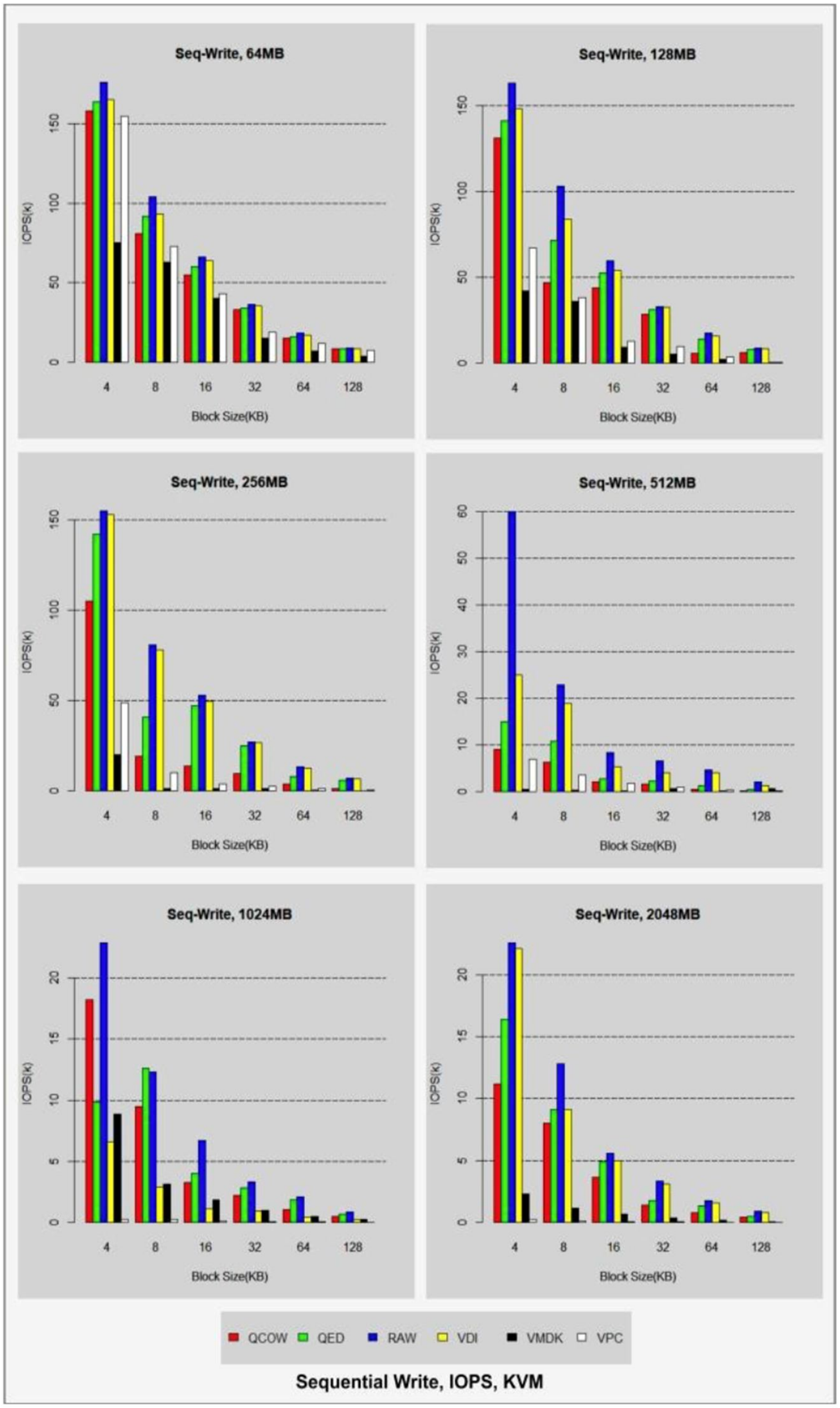

Fig. 17. IOPS parameter analysis for sequential write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 17 shows the IOPS for different disk image file formats for sequential write input on KVM hypervisor. RAW and VDI virtual disk perform well for all file sizes and on all block sizes. For small file sizes, all virtual disk images perform well. As the file size or block size increases, the performance of all virtual disk image file formats decreases. With an increase in file size, VMDK virtual disk's performance increases, whereas the performance of VPC decreases. The overall preferable sequence of virtual disk image file formats for a sequential write based on IOPS, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Fig. 18 shows the IOPS for different disk image file formats for sequential write input on VirtualBox hypervisor. VMDK and VDI perform well for all file sizes and block sizes, whereas VHD and HDD perform worst for small file sizes and for large file sizes QCOW and QED perform the worst. With an increase in the file size, QCOW virtual disk image's performance decreases, whereas the performance of VHD virtual disk increases. The overall preferable sequence of virtual disk image file formats for a sequential write based on IOPS, on VirtualBox hypervisor is VMDK, VDI, HDD, VHD, QED and QCOW.

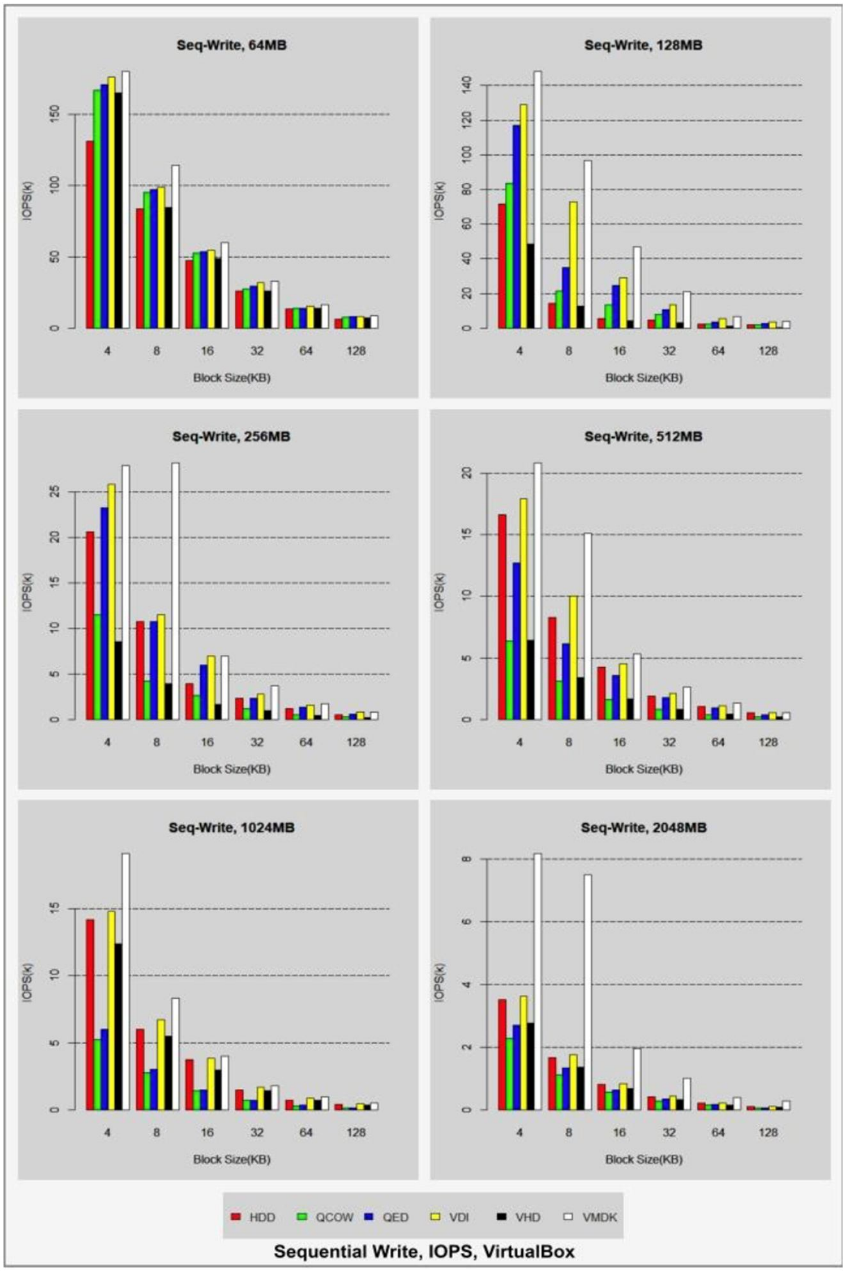

Fig. 18. IOPS parameter analysis for sequential write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for sequential read pattern, we can infer that disk image file formats which perform well have IOPS of 20k and above, those with average performance have IOPS between 10k and 20k, whereas disk image file formats which perform poorly have IOPS of 10k and less. The overall preferable disk image file format for a sequential write operation based on IOPS on both hypervisors is VDI. Among both hypervisors, VMDK performs best on VirtualBox hypervisor and RAW performs best on KVM hypervisor.

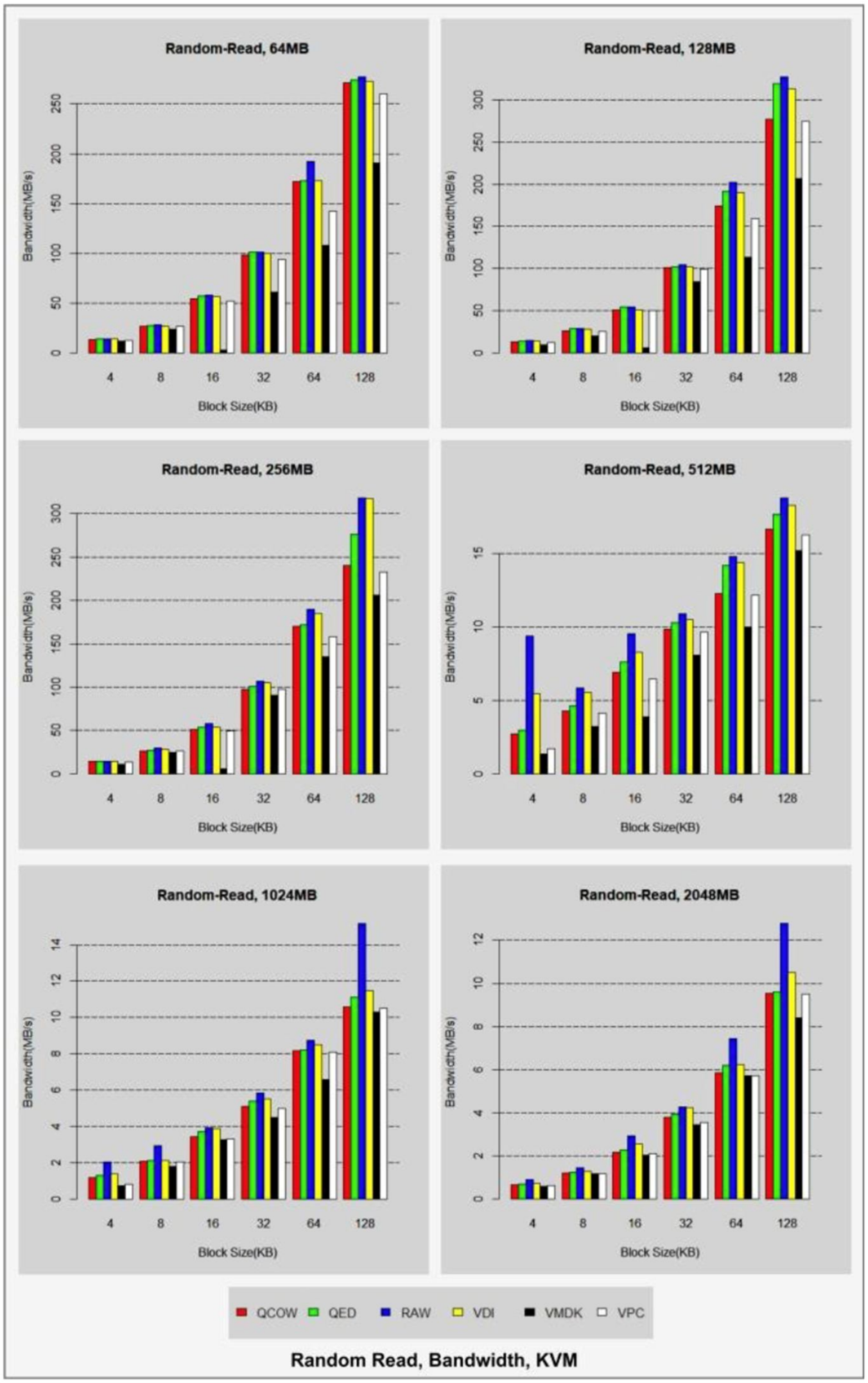

Fig. 19. Bandwidth parameter analysis for random read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 19 shows the bandwidths for different disk image file formats for random read input on KVM hypervisor. RAW and VDI virtual disk perform well for all file sizes and for all block sizes. For small file size, all virtual disk images perform well but as the file size increases, the performance of all virtual disk images decrease. With an increase in the file size, the performance of VMDK virtual disk increases, whereas the performance of VPC decreases. The overall preferable sequence of virtual disk image file formats for a random read operation based on bandwidth, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Fig. 20 shows the bandwidths for different disk image file formats for random read input on VirtualBox hypervisor. VMDK and VDI perform well for all file and block sizes, whereas VHD and HDD perform the worst. With an increase in the file size, QCOW disk image's performance decreases, whereas the performance of HDD disk image increases. As block size increases, HDD file format's performance increases. The overall preferable sequence of virtual disk image file formats for a random read operation based on bandwidth, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

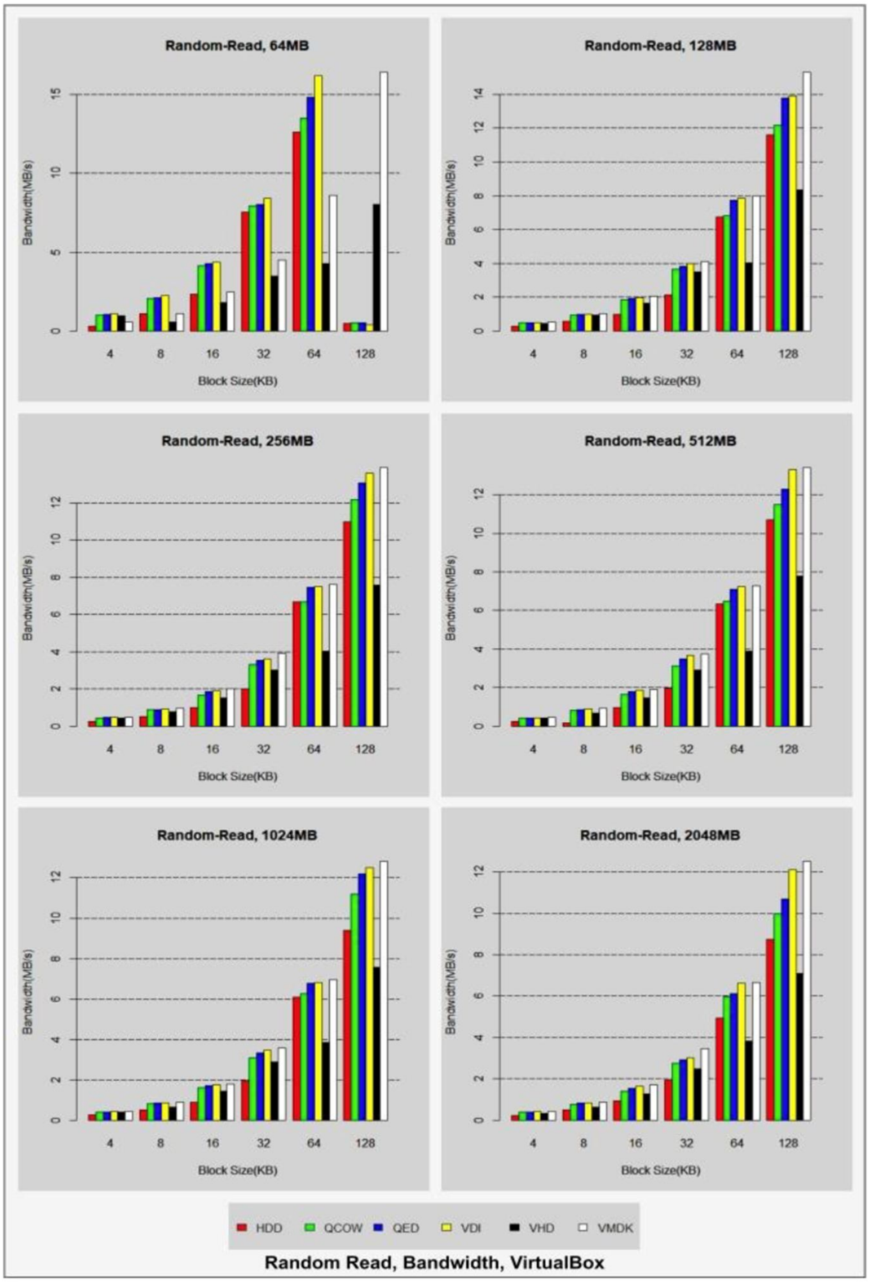

Fig. 20. Bandwidth parameter analysis for random read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for random read operation, we can infer that disk image file formats which perform well have bandwidth of 8 MB/s and above, those with average performance have bandwidth ranging from 6 to 7 MB/s, whereas disk image file formats which perform poorly have bandwidth of 6 MB/s and less. The overall preferable disk image for a random read based on bandwidth on both hypervisors is VDI. Among both hypervisors, VMDK performs best on VirtualBox hypervisor, but it performs the worst on KVM hypervisor.

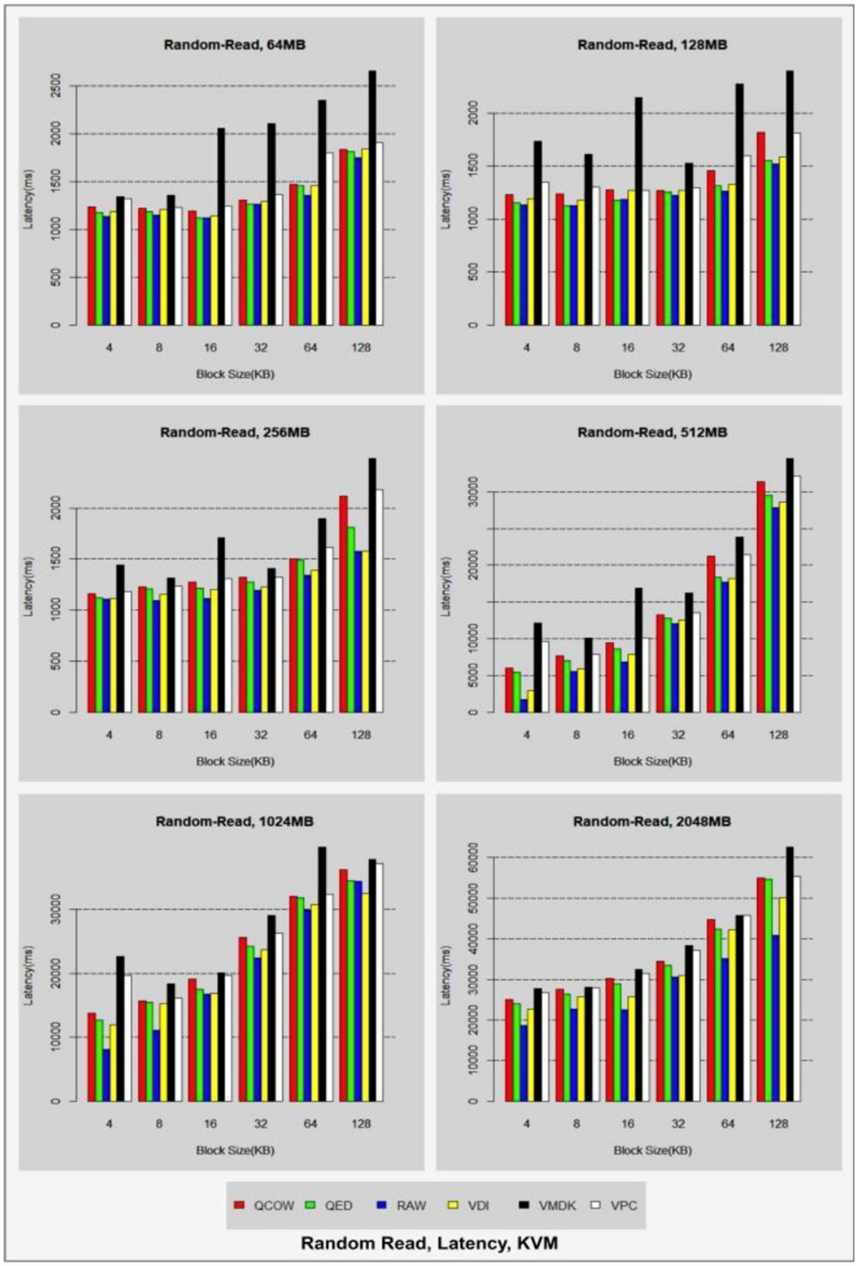

Fig. 21. Latency parameter analysis for random read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 21 shows the latencies for different disk image file formats for random read input on KVM hypervisor. RAW and VDI disk formats perform well for all file and block sizes. For small file sizes, all virtual disk images perform well and the performance decreases with increase in block size as expected. With an increase in the file size, the performance of majority of virtual disk image file formats decreases. As the file size increases, VMDK's relative performance increases, whereas the VPC file format's performance decreases. The overall preferable sequence of virtual disk image file formats for a random read based on latency, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

Fig. 22 shows the latencies for different disk image file formats for random read input on VirtualBox hypervisor. VMDK and VDI perform well for all file and block sizes, whereas VHD and HDD perform the worst. With an increase in the file size, the performance of QCOW in comparison to other file formats decreases. As the block size increases, HDD format's performance increases. The overall preferable sequence of virtual disk image file formats for a random read based on latency, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

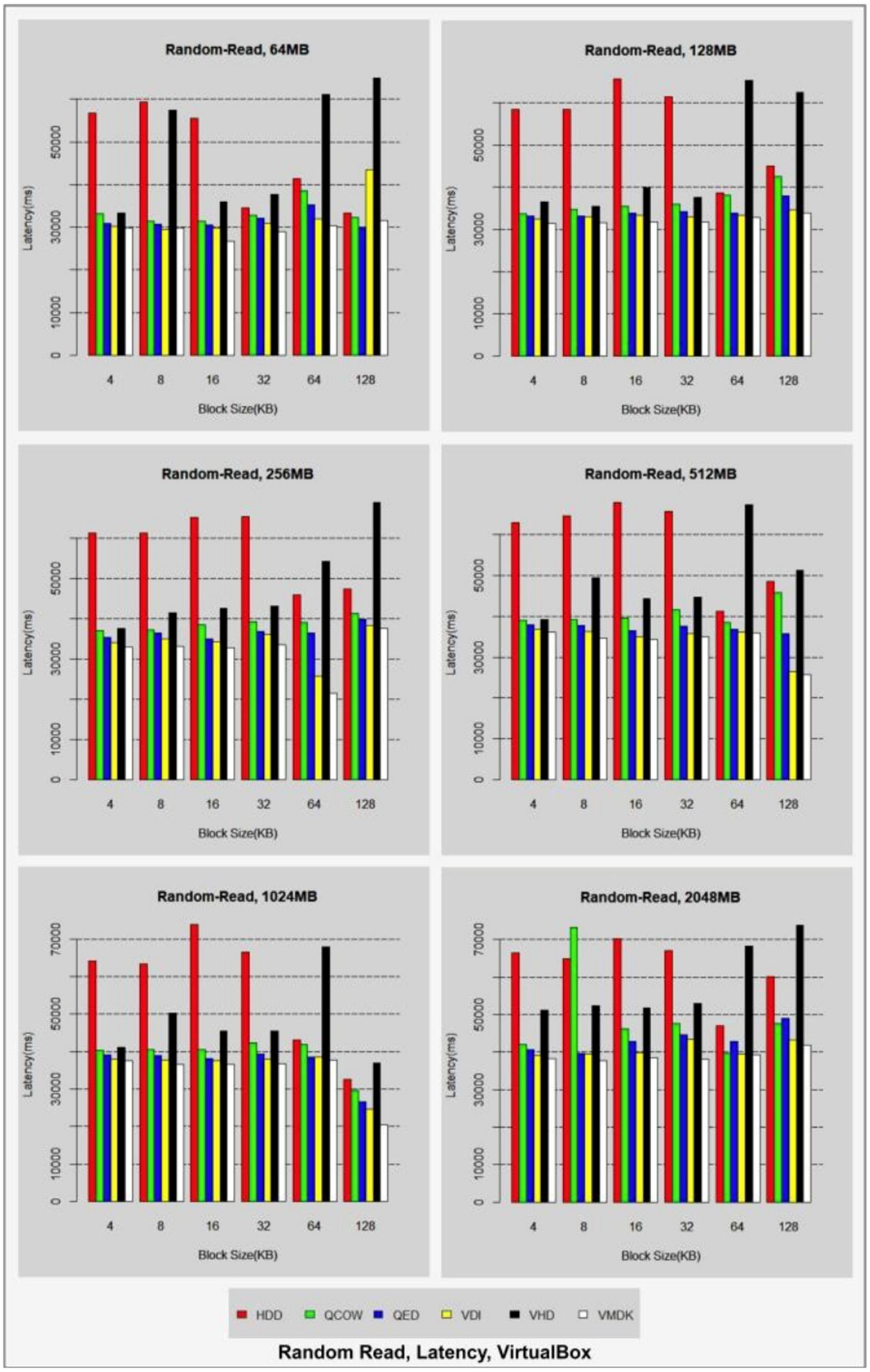

Fig. 22. Latency parameter analysis for random read on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for random read operation, we can infer that disk image file formats which perform well have latency of 10000 ms and less, those with average performance have latency between 10000 and 30000 ms, whereas disk image file formats which perform poorly have latency of 30000 ms and above. The overall preferable virtual disk image format for a random read based on latency, on both hypervisors is VDI. Among both hypervisors, VMDK performs best on VirtualBox hypervisor, although VMDK performs worst on KVM hypervisor. It is recommended to use KVM hypervisor for small file sizes because it provides lower latency in comparison to VirtualBox hypervisor.

Fig. 23 shows the IOPS for different disk image file formats for random read input on KVM hypervisor. RAW and VDI virtual disk formats perform well for all file and block sizes. For small file sizes, all virtual disk images perform well, but as the file size increases, all virtual disk images see a decrease in performance. With the increase in file sizes, the VMDK virtual disk's performance increases, whereas the performance of VPC decreases. The overall preferable sequence of virtual disk image file formats for a random read based on IOPS, on KVM hypervisor is RAW, VDI, QED, QCOW, VPC and VMDK.

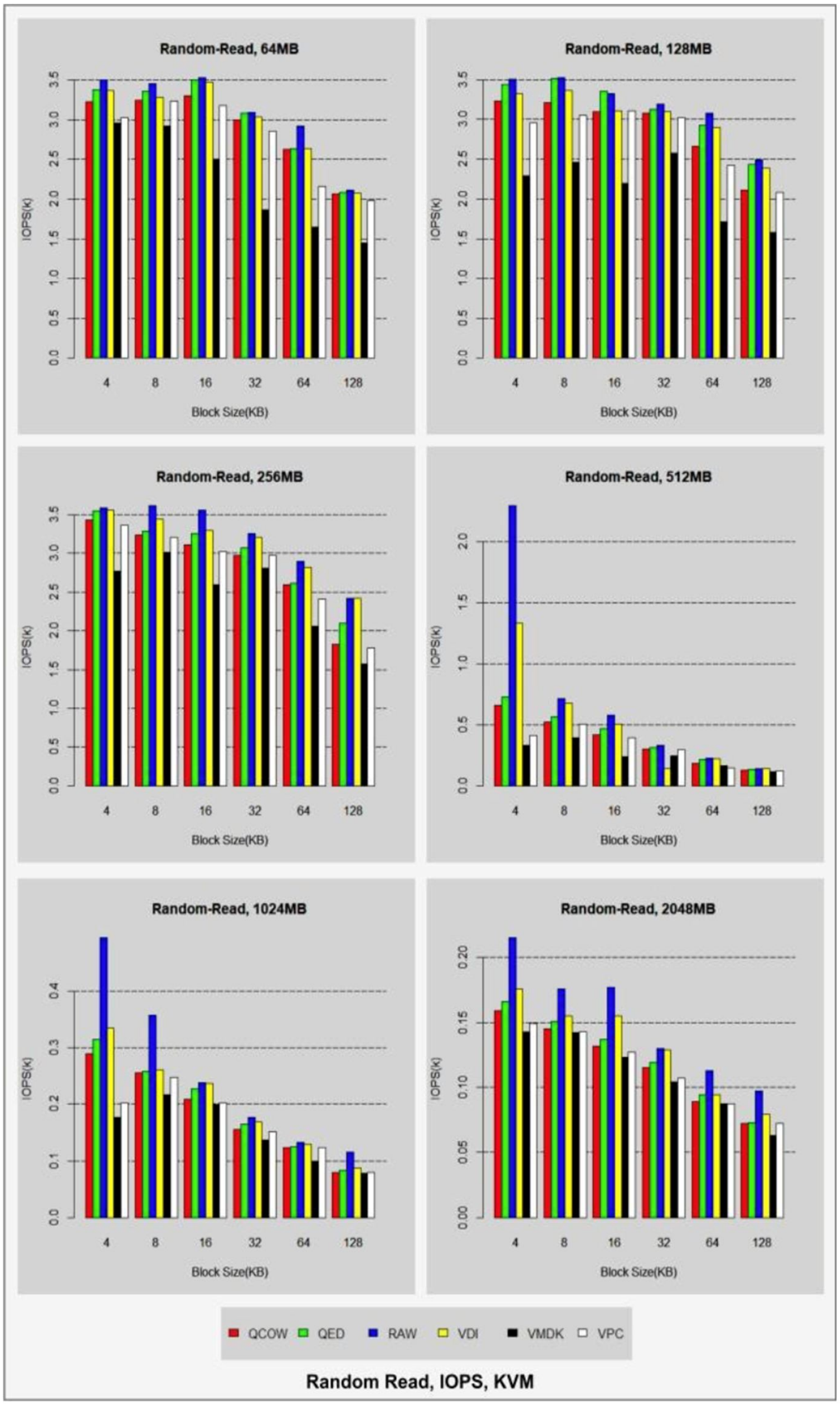

Fig. 23. IOPS parameter analysis for random read on block sizes 4KB-128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Fig. 24 shows the IOPS for different disk image file formats for random read input on VirtualBox hypervisor. VMDK and VDI perform well for all file and block sizes, whereas VHD and HDD perform the worst. With an increase in the file size, QCOW virtual disk image's performance decreases, whereas the HDD virtual disk image's performance increases. As the block size increases, HDD format's performance increases. The overall preferable sequence of virtual disk image file formats for a random read based on IOPS, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

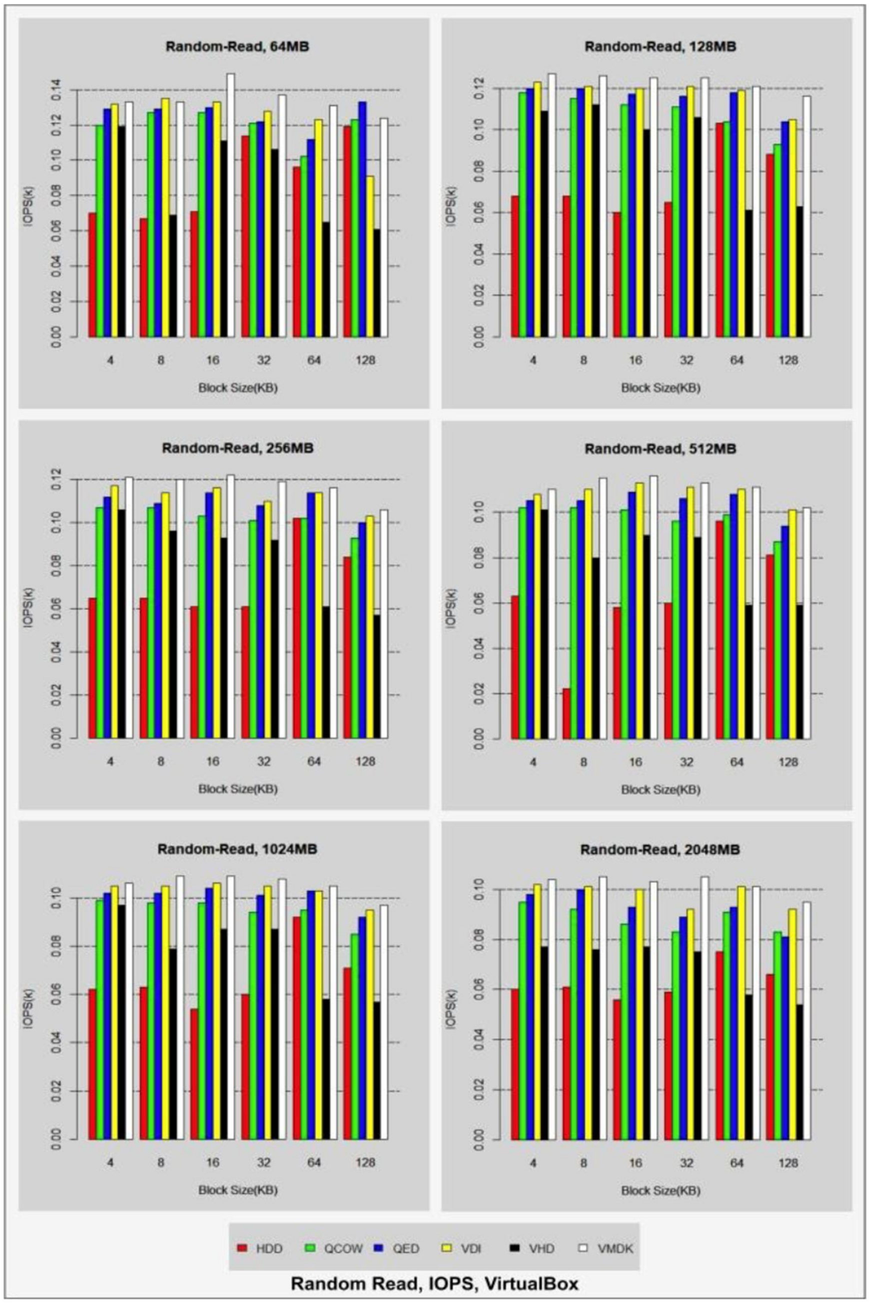

Fig. 24. IOPS parameter analysis for random read on block sizes 4KB-128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for random read pattern, we can infer that disk image file formats which perform well have IOPS of 0.55k and above, those with average performance have IOPS between 0.2k and 0.55k, whereas disk image file formats which perform poorly have IOPS of 0.2k and less. The overall preferable virtual disk image file formats for a random read operation based on IOPS, on both hypervisors is VDI. Among both hypervisors, VMDK performs best on VirtualBox hypervisor, although it performs the worst on KVM hypervisor.

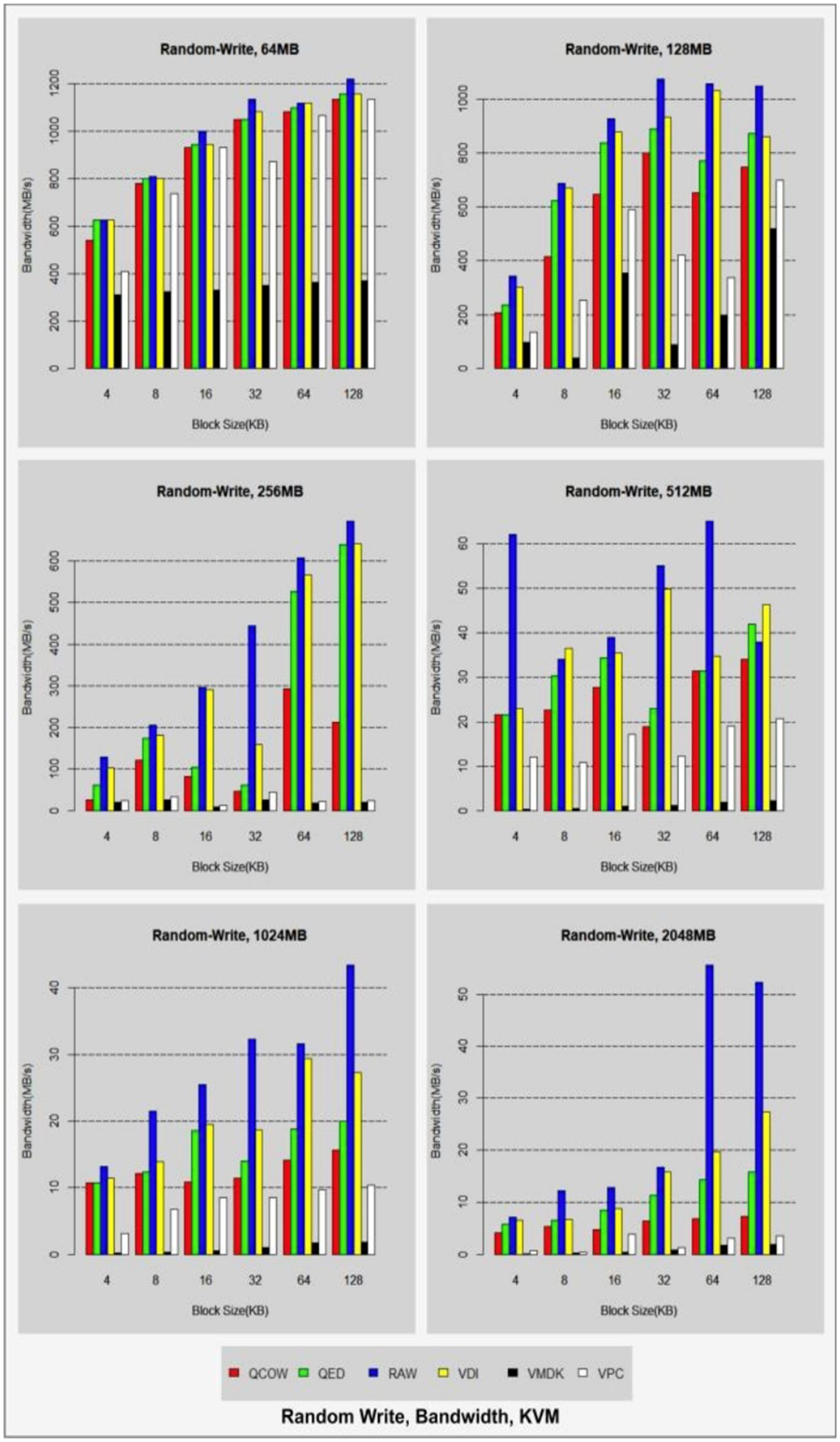

Fig. 25. Bandwidth parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 25 shows the bandwidths for different disk image file formats for random write input on KVM hypervisor. RAW and QED virtual disks perform well for all file and block sizes. For small file sizes, all virtual disk images perform well. With an increase in the file size, the performance of all virtual disk images decrease. The overall preferable sequence of virtual disk file formats for a random write operation based on bandwidth, on KVM hypervisor is RAW, QED, VDI, QCOW, VPC and VMDK.

Fig. 26 shows the bandwidths for different disk image file formats for random write input on VirtualBox hypervisor. VMDK and VDI perform well for all file and block sizes, whereas HDD and VHD perform the worst. As the block size increases, QCOW file format's performance increases, whereas the performance of QED decreases. For file sizes: 1024MB and 2048 MB, HDD performs better than VHD on block sizes: 64KB and 128KB. The overall preferable sequence of virtual disk image file formats for a random write operation based on bandwidth, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

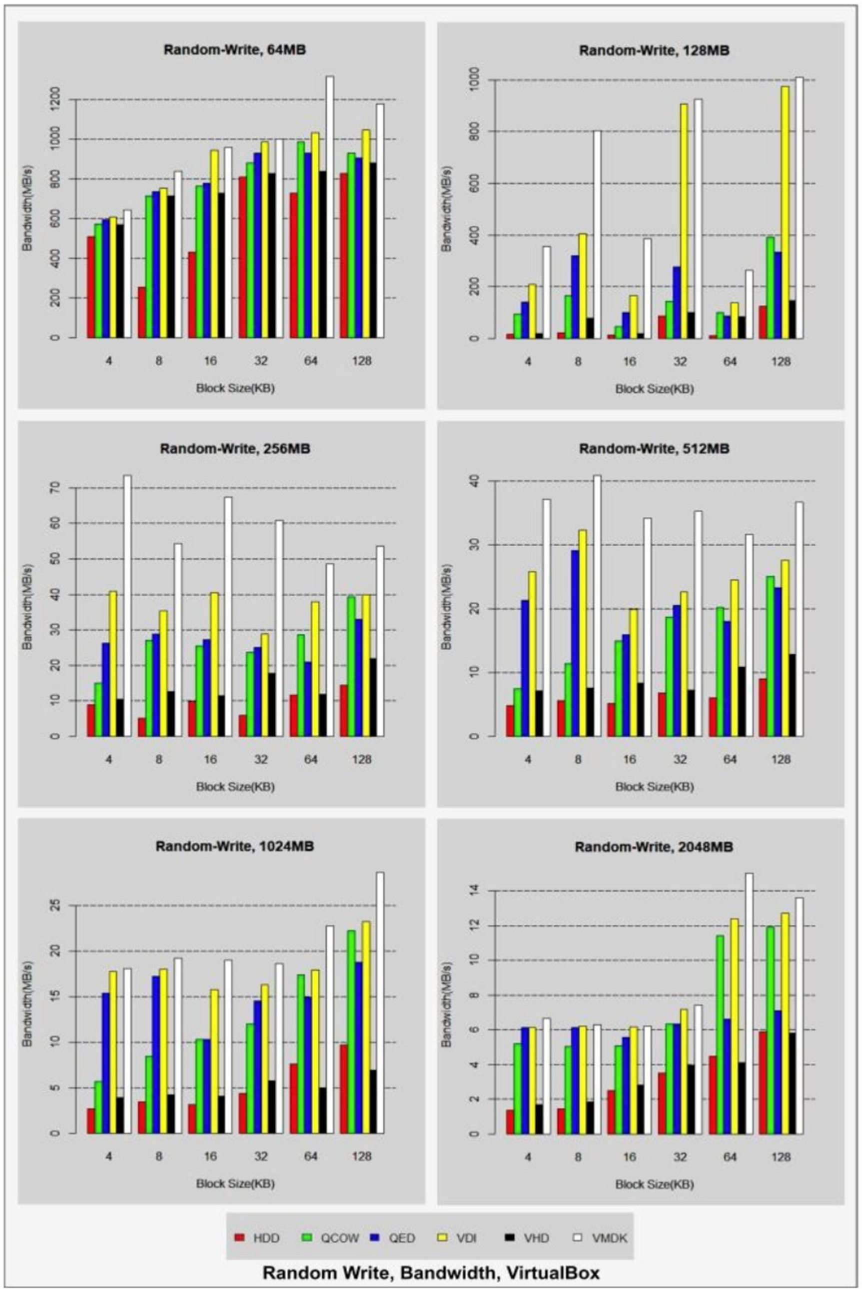

Fig. 26. Bandwidth parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB VirtualBox hypervisor.

Based on above results for random write IO operation, we can infer that disk image file formats which perform well have bandwidth of 20 MB/s and above, those with average performance have bandwidth ranging from 10 to 20 MB/s, whereas disk image file formats which perform poorly have bandwidth of 10 MB/s and less. The overall preferable virtual disk image file format for a random write operation based on bandwidth, on both hypervisors is VDI. VMDK has the best performance on VirtualBox hypervisor while it performs the worst on KVM hypervisor.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

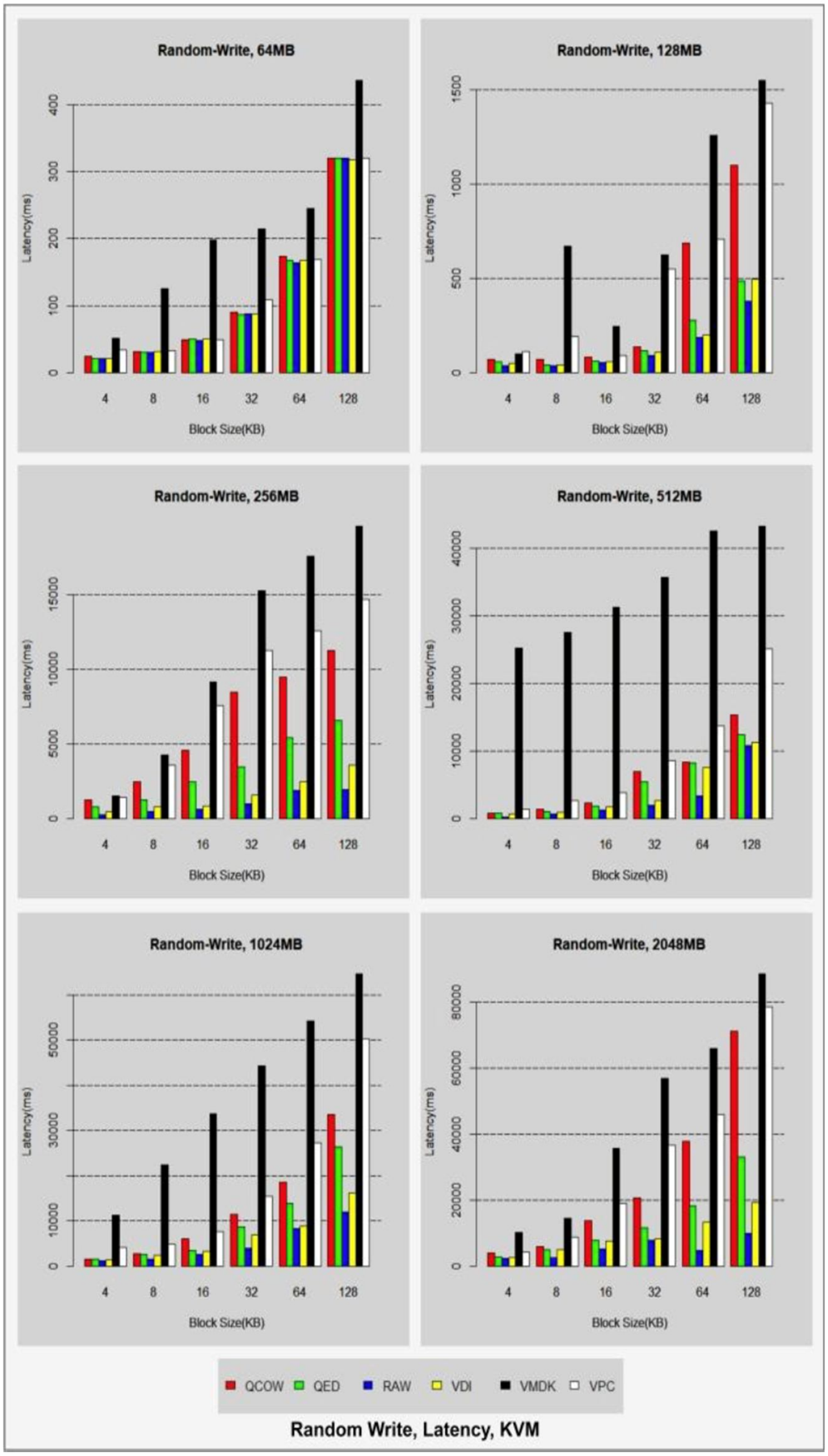

Fig. 27. Latency parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 27 shows the latencies for different disk image file formats for random write input on KVM hypervisor. RAW and QED virtual disk formats perform well for all file and block sizes. For small file sizes, all virtual disk images perform well but as the file sizes increase, the performance of all virtual disk images decrease. The overall preferable sequence of virtual disk image file formats for a random write based on latency, on KVM hypervisor is RAW, QED, VDI, QCOW, VPC and VMDK.

Fig. 28 shows the latencies for different disk image file formats for random write input on VirtualBox hypervisor. VMDK and VDI formats perform well for all file and block sizes, whereas VHD and HDD perform the worst. With an increase in the block size, QCOW virtual disk image's performance increases and the performance of QED decreases. For file sizes: 1024MB and 2048 MB and block sizes: 64 and 128 KB, HDD performs better than VHD format. The overall preferable sequence of virtual disk image file formats for a random write operation based on latency, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

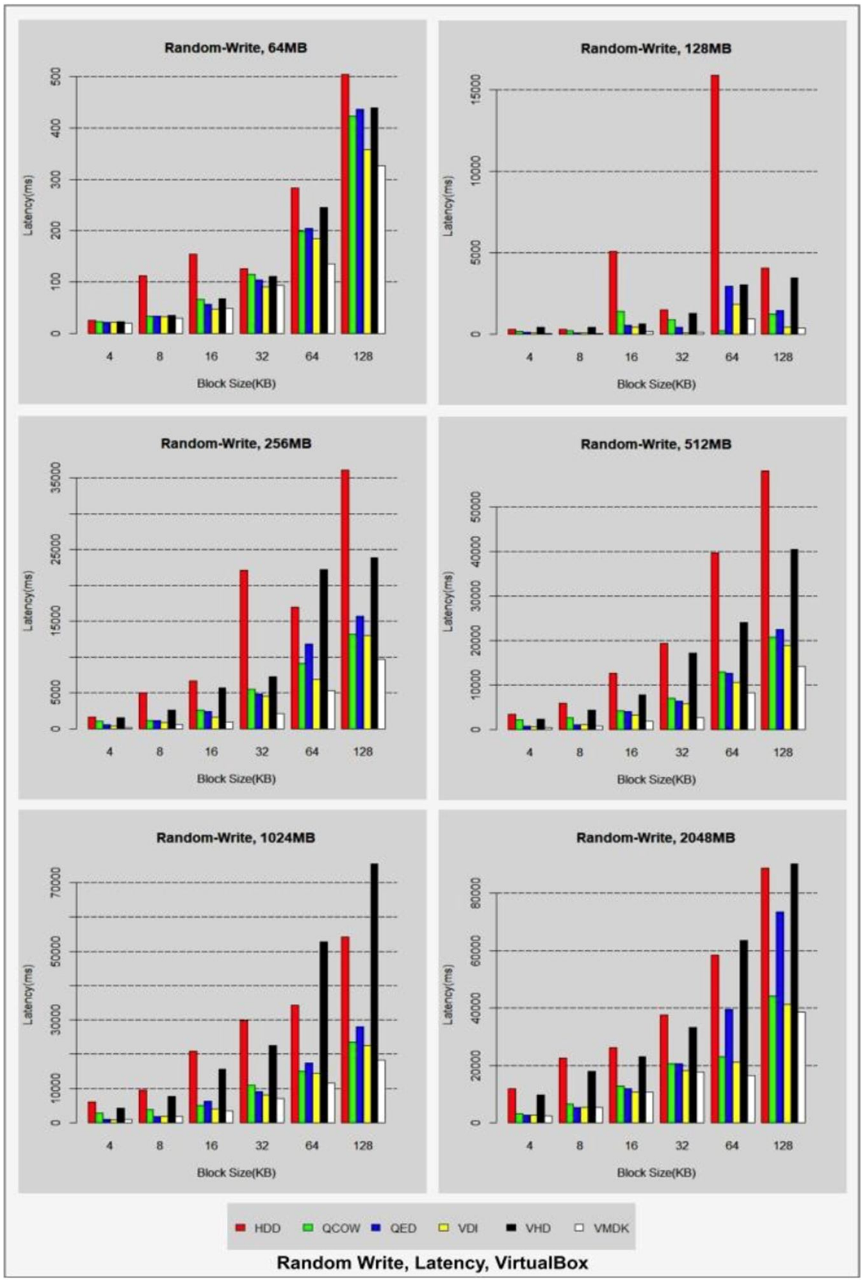

Fig. 28. Latency parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for random write IO operation, we can infer that disk image file formats which perform well have latency of 20000 ms and less, those with average performance have latency between 20000 and 40000 ms, whereas disk image file formats which perform poorly have latency of 40000 ms and above. The overall preferable virtual disk image file format for a random write operation based on latency, on both hypervisors is VDI. VMDK performs the best on VirtualBox hypervisor while it performs the worst on KVM hypervisor.

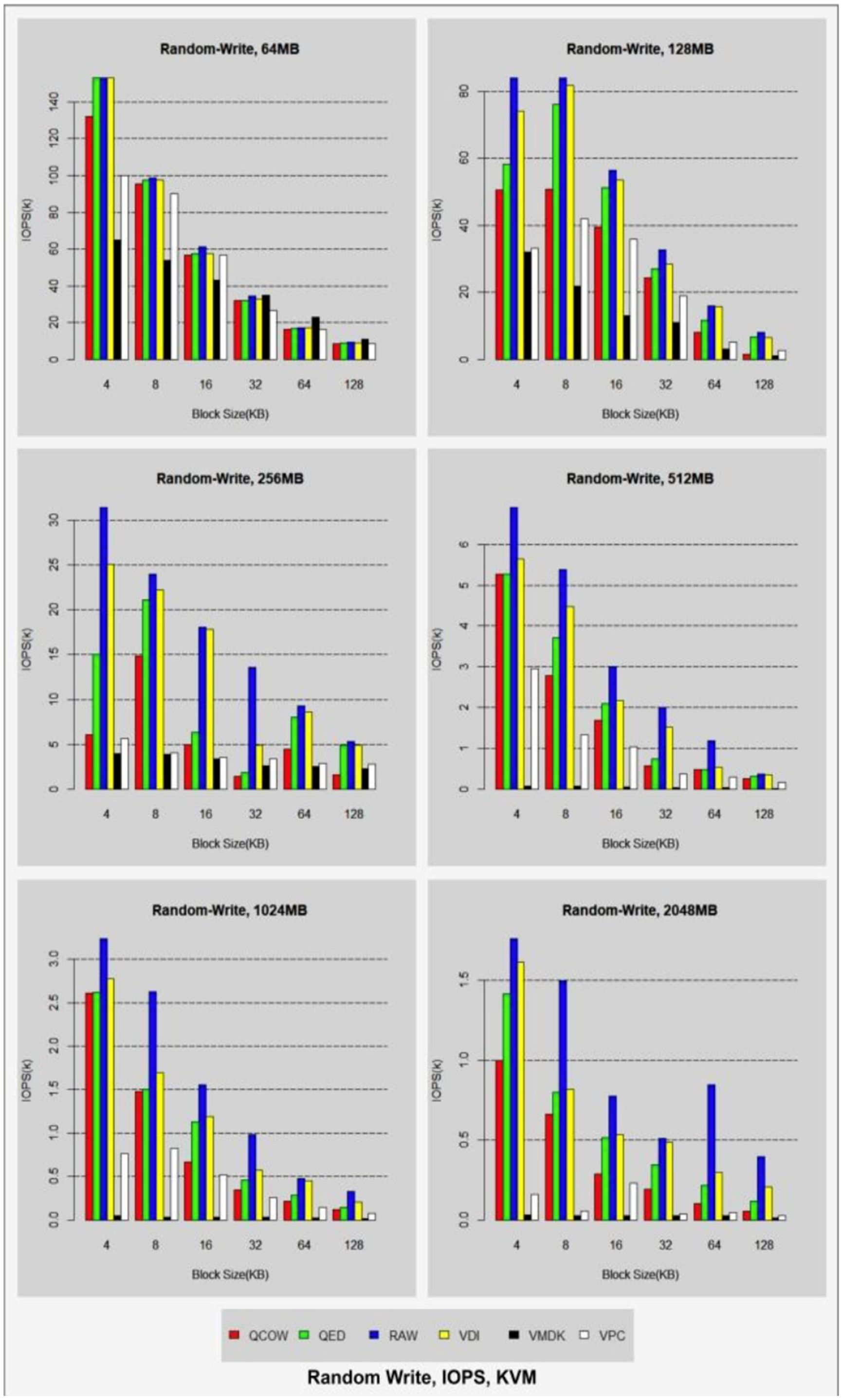

Fig. 29. IOPS parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on KVM hypervisor.

Fig. 29 shows the IOPS for different disk image file formats for random write input on KVM hypervisor. RAW and QED virtual disk image formats perform well for all file and block sizes. For small file sizes, all virtual disk images perform well but as the file size increased, all virtual disk images saw a decrease in their performance. The performance of VMDK virtual disk decreases drastically as the file size increases. The overall preferable sequence of virtual disk image file formats for a random write operation based on IOPS, on KVM hypervisor is RAW, QED, VDI, QCOW, VPC and VMDK.

Fig. 30 shows the IOPS for different disk image file formats for random write input on VirtualBox hypervisor. VMDK and VDI formats perform well for all file size and block size, whereas VHD and HDD perform the worst. For file sizes: 1024MB and 2048 MB and block sizes: 64KB and 128KB, HDD performs better than VHD. The overall preferable sequence of virtual disk image file formats for a random write operation based on IOPS, on VirtualBox hypervisor is VMDK, VDI, QED, QCOW, VHD and HDD.

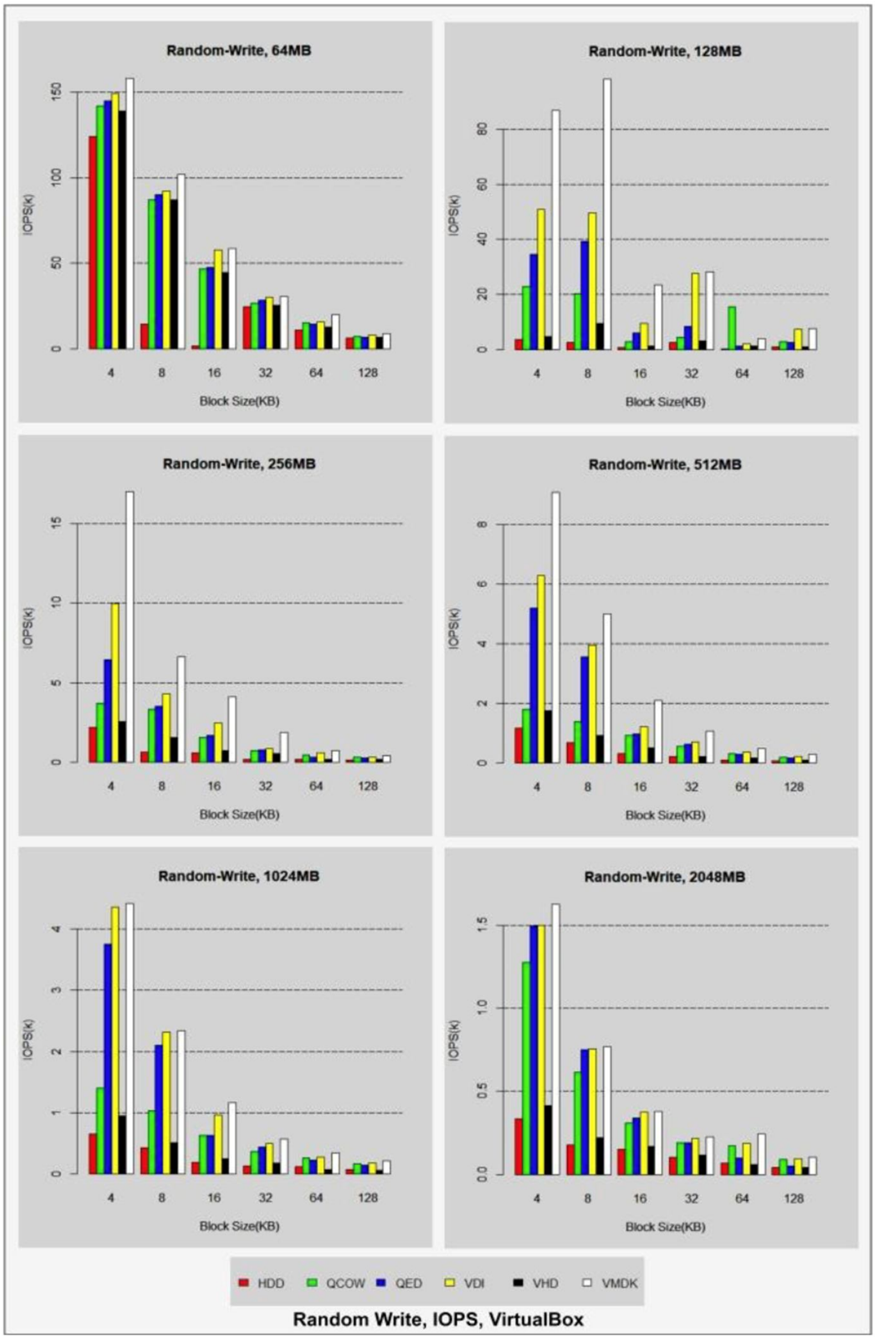

Fig. 30. IOPS parameter analysis for random write on block sizes 4KB- 128KB for different file sizes 64MB – 2048MB on VirtualBox hypervisor.

Based on above results for random write operation, we can infer that disk image file formats which perform well have IOPS of 4k and above, those with average performance have IOPS between 2k and 4k, whereas disk image file formats which perform poorly have IOPS of 2k and less. The overall preferable virtual disk image file format for a random write operation based on IOPS on both hypervisors is VDI. VMDK performs best on VirtualBox hypervisor and VMDK perform worst on KVM hypervisor. It is recommended to use VirtualBox hypervisor for all file sizes since it provides a high IOPS value in comparison to VirtualBox.

B. Power Consumption Result

Taking Congfeng Jiang et al [14] as reference we study the power consumption of different formats on the different hypervisors. Fig. 31 provides the power consumption in watts for different virtual disk image file formats running on KVM hypervisor. Virtual disk file formats are labelled on the x-axis and power consumption in watts is present on the y-axis. This power consumption is captured using the same workload for random read operation with file size of 1024MB and block size of 128KB. The background processes running on KVM hypervisor are ignored for the study. The results indicate that VPC and VMDK formats draw more power than other virtual disk file formats due to their lower performance on KVM and high latencies whereas, RAW and QCOW formats consume lesser power in comparison to other virtual disk file formats owing to better performance and low latencies on KVM. The overall preferable sequence of virtual disk image file formats from a power consumption perspective on KVM hypervisor is RAW, QCOW, QED, VDI, VMDK and VPC.

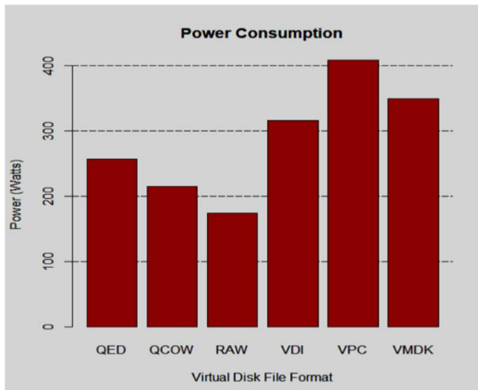

Fig. 31. Watts parameter analysis for different virtual disk file format on KVM.

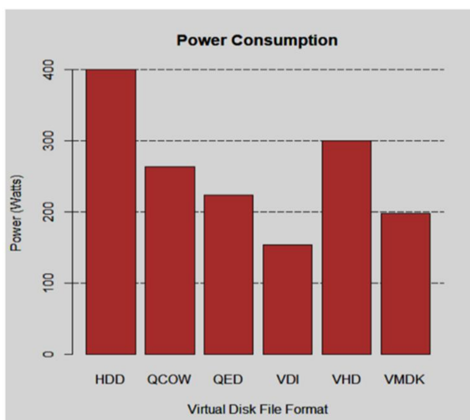

Fig. 32. Watts parameter analysis for different virtual disk file format on VirtualBox.

Fig. 32 provides the power consumption in watts for different virtual disk image file formats running on VirtualBox hypervisor. Virtual disk file formats are labelled on the x-axis and power consumption in watts is present on the y-axis. This power consumption is captured using the same workload for random read operation with file size of 1024MB and block size of 128KB. The workload is the same as that which was used on KVM hypervisor. The background processes running on VirtualBox are ignored in the study. The results indicate that HDD and VHD formats consume more power than other virtual disk file formats due to their lower performance and high latencies, whereas VDI consumes lesser power than other virtual disk file format due to its good performance and low latency time. QED, QCOW and VMDK consume average power consumption on VirtualBox hypervisor. The overall preferable sequence of virtual disk image file formats from a power consumption perspective on VirtualBox hypervisor is VDI, VMDK, QED, QCOW, VHD and HDD.

In general, virtual disks on KVM hypervisor consume more power (watts) in comparison to VirtualBox. Hence it is recommended to use the VirtualBox hypervisor for better power efficiency.

V. CONCLUSION

An empirical study of the performance for different types of virtual disk image file formats when executed on KVM and VirtualBox hypervisors was presented in this research paper. To perform a comprehensive and comparative study four different parameters namely: latency, bandwidth, IOPS and power consumption (in watts) were used. For performance study, we have used the fio benchmark and for power consumption study fio and powerstat tools were used. This study offers guidelines to virtual disk end users for choosing the appropriate virtual disk image file format for their use case. It will prove to be a valuable reference for developers while building new virtual hard disk or modifying any existing one. This study also offers to choose the best hypervisor based on the use case of the user.

VI. FUTURE SCOPE

In future, we plan to increase the scope of this study even more. Firstly, we plan to incorporate and use different I/O benchmark tools like IOzone for the performance study. Secondly, we plan to include other hypervisors such as XEN and Microsoft's Hyper-V. Thirdly, we plan to perform this study on concurrently working VMs and on multiple machines with a different configuration. Furthermore, we plan to use many different and varied IO operations. Finally, we plan to perform exhaustive analysis to find the root cause of the apparent runtime performance of different virtual disk based on how they are created.

REFERENCES

[1] Virtualization, See https://www.webopedia.com/virtualization.html, 25 June 2018.
[2] Be aware of the following limitations n a virtual, See https://www.coursehero.com/file/p1p04sik/Be-aware-of-the- following-limitations-n-A-virtual-machine-cannot-have-more/, 25 June 2018.
[3] L. YamunaDevi, P. Aruna, Devi D. Sudha and N. Priya, "Security in Virtual Machine Live Migration for KVM", Process Automation, Control and Computing (PACC), 2011 International Conference, 2011.
[4] Kritwara Rattanaopas and Pichaya Tandayya, "Comparison of disk I/O power consumption in modern virtualization", Computer, Communications, and Control Technology (I4CT), 2015.
[5] What are Virtualization and Cloud, See https://www.quora.com/What-is-virtualization, 25 June 2018.
[6] Alexandru-Catalin Bujor and Razvan Dobre, "KVM IO Profiling", Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, 2013.
[7] Jianhua Che, Qinming He, Qinghua Gao and Dawei Huang, "Per- formance Measuring and Comparing of Virtual Machine Monitors", IEEE/IFIP International Conference, 2008.
[8] Stefan Gabriel Soriga and Mihai Barbulescu, "A comparison of the performance and scalability of Xen and KVM hypervisors", Networking in Education and Research, RoEduNet International Conference 12th Edition, 2013.
[9] Ca ̆ta ̆lina Mancas ̧, "Performance improvement through Virtualization", RoEduNet International Conference - Networking in Education and Research, 2015.
[10] Flexible I/O, See https://fio.readthedocs.io/en/latest/fio doc.html, 25 June 2018.
[11] Power Consumption Calculator for Ubuntu Linux, See http://manpages.ubuntu.com/manpages/trusty/man8/powerstat.8.html, 25 June 2018.
[12] Virtual Machine Manager, See https://virt-manager.org/, 25 June 2018.
[13] Gauri Joshi, S.T. Shingade and M.R. Shirole, "Empirical Study of Virtual Disks Performance with KVM on DAS", IEEE International Conference, 2014.
[14] Congfeng Jiang, Dongyang Ou, Yumei Wang, Energy Efficiency Comparison of hypervisors.