Windows Server 2008

Windows Server 2008 R2

Virtual Hard Disk Performance

Windows Server 2008 / Windows Server 2008 R2 / Windows 7

Authors:
Liang Yang
Anthony F. Voellm

A Microsoft White Paper
Published: March2010

Contents

1. Introduction

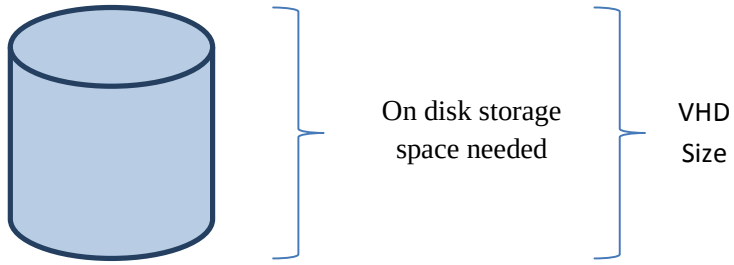
A virtual hard disk (VHD) is a file that encapsulates a hard disk image. VHDs can be used in new and interesting ways. VHDs first were created to be the storage media for virtual machines (VMs). Today, VHDs are used to ship trial versions of software, used in backup solutions, used for bug triage (e.g. customers can convert a physical disk to virtual and share it), and even used to store multiple boot environments.VHDs are a very flexible storage container and are not tied to any single file system format. Since June 2005, Microsoft has made the VHD Image Format Specification available to third parties under the Microsoft Open Specification Promise (OSP).

Microsoft began using VHD technology in Microsoft Virtual PC around 2003, and then continued its use in Microsoft Virtual Server release in 2005.The next major release happened as part of Hyper-V in Windows Server 2008. Currently VHD supportis made as part of Windows Server 2008 R2 and high-end client SKUs of Windows 7. VHDswerelimited to use by virtual machines running in Virtual PC/VirtualServer/Hyper-V and loopback mounting of VHDs in the parentpartition sometimes referred to as the management operating system.The integration of VHD support into the operating system was drastically improved in Windows Server 2008 R2 which added native support. Native support means the technology is integrated into the operating system and no longer requires a virtualization solution such as Hyper-V to be available. Native support added the following features:

- boot from VHD
- integrated support for attaching and detaching VHDs via inbox APIs (virtdisk.dll)and the "Disk Management" control panel (diskmgmt.msc or command line tool DiskPart ("attaching" is the term used to describe the action of mounting a VHD so it can be used by Windows directly, or as a disk in the guest operating system of a virtual machine).
- attachingVHDs from inside VHDs, and many performance improvements.

There are three VHD formats each with different performance characteristics. The three formats of VHD are fixed, dynamic and differencing.

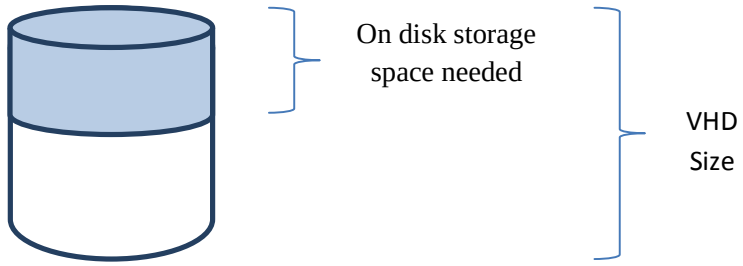
Fixed Sized Virtual Hard Disk



A fixed sized VHDuses a file in which the space to store the file is allocated on the physical storage when the virtual hard disk is created. The file size is the same as the size specified for the virtual hard disk. As their name implies, fixed sized VHDs occupy the same space on the underlying physical storage device as their specified size. However,

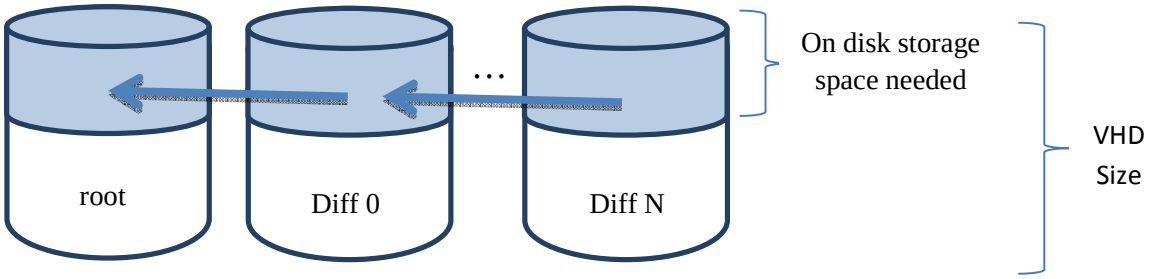
once a fixed sized VHD is created,the size can be increased when the disk is offline by editing the disk to expand it.Reducing the size is not supported. Because the physical storage required for a fixed size VHD is allocated when the VHD is created, there is a better chance at optimal placement and organization on-disk which yields the best performance. The disadvantage is the space is committed even if it is not used.

Dynamically Expanding Virtual Hard Disk



A dynamically expandingVHD is a file that at any given time is as large as the actual data written to it plus the size of on-disk meta-data. Dynamically expanding disks are useful because they do not require all the storage needed to contain the maximum size of the disk to be reserved up front. The VHD filestarts quite small (e.g. 42KB is a typical physical size of an empty 20GB disk) and grows as new blocks in the disk are used. There are a number of optimizations around dynamically expanding disks that improve performance; however, in general their read/write performance is slower than fixed disks. One optimization is the selection of data block size which can be either 512KBor 2MB; another is skipping allocation of all-zero blocks.

Differencing Virtual Hard Disk



A differencingVHD is a file representing the current state of the virtual hard disk as a set of modified blocks in comparison to a parent virtual hard disk. DifferencingVHDs can be associated with either a fixed sized or dynamically expanding VHD. DifferencingVHDs can also be associated withanother differencing VHD but they cannot be associated with a physical disk.Differencing VHDs are used to prevent changes from being made in theirparent VHD to which they are applied and are used to implement a number of additional features. In Hyper-V, differencing VHDs are also created

automatically whenever snapshots are taken of a virtual machine. Note differencing VHDs used for snapshot purpose are named with an AVHD file extension to help users easily distinguish them from regular differencing VHDs. Differencing VHDs may alsobe used to deploy a “golden” or “master” image, because you can associate multiple differencing VHDs to one parent VHD. Some disadvantages of differencing VHDs are increased caching needs and the inability to grow or shrink the VHD size. You can however compact differencing VHDs to reclaim physical space usage.

There are several important limitations for VHDs:

- VHDs can be mounted only on NTFSvolumes (although you can still save a VHD file on FAT/FAT32 assuming the maximum file size limit is not violated). For example, if you have a differencing VHD chain, then every VHD along the chain must sit on an NTFSvolume to make VHD attaching work.
- VHDs cannot be mounted within a compressed folder in Windows Server 2008 R2.This was possible in Hyper-V role in Windows Server 2008, but this capability was explicitly blocked in the Hyper-V role in Windows Server 2008 R2since the compressed file size limitis relatively small. A dynamically expandable VHD can easily outgrow that limit and getcorrupted.
- In addition to the maximum file size of NTFS, dynamic or difference VHDs cannot exceed 2040GB.The reason for the 2040G limit is the length of each Block Allocation Table entry is set to 4 Bytes and the maximum valid value is 0xFFFFFFFE (0xFFFFFFFF means an unused entry). If you multiply that value by 512B sector size and then subtract the overhead of on disk meta-data structures, 2040G will be the maximum size of dynamically or differencing VHDs.

The remainder of this paper is focused on giving in-depth performance data on Virtual Hard Disks. Comparisons are generally made between native, Window Server 2008 and Windows Server 2008 R2. When comparing Windows Server 2008 and Window Server 2008 R2, no virtual machines are used unless explicitly stated. The goal of this paper is to look at virtual hard disk performance and not Hyper-V performance.

2. Performance Measurement Methodologies

A. Test Details

Storage performance testing is the science used to determine the characteristics of storage I/O subsystem. The performance results are always related to certain software and hardware platform settings within a controlled experimental environment. We take into consideration the following factorsand assumptions to make meaningful VHD performancemeasurements and comparisonswhich are crucial to obtain valid conclusions.

Choice of Targets: Raw Disk vs. Regular File vs. VHD

We use raw disk (aka physical disk or bare-metal disk) and raw file (i.e. a normal non-VHD file) to compare with native VHD performanceresults. The benefits of adding these two baselines can help us quickly determine if a performance issue or performance improvement is solely VHD-related or comes from other Windows components.

- A raw diskprovides direct access to the storage media instead of through its file system andis considered the thinnest layer in terms of sending I/Os down to the disk. This can give us a good estimation of what the maximum throughput and minimum CPU cycles a VHD can achieve. Occasionally a raw disk does not give the best performance due to additional optimization (e.g. I/O batching) done by other layers, but so far it has been proved a sound guard for sanity validation purposes.
- A raw file shares thecommon I/O paths with a VHD file except those added by the VHD driver stack. By comparing a VHD with a raw file, we can find out how much overhead theVHD adds on top of a raw file. Ideally we want a VHD performance to be on-par witha raw filefor the major performance metrics.

Windows hypervisor impact on performance measurement

In Windows Server 2008 R2, VHD support is nativeto the operating system and does not depend on the presence of the Windows hypervisor. From a throughput perspective, the impact of the Windows hypervisor is quite small based on past experimentalresults. This is mainly due to the fact that performance critical workloads are re-directed to synthetic VMBus channels instead of using the longer emulation path. To get the most accurate CPU utilization and to focus on native performance, the Windows hypervisor is turned on only during VHD performance measurement in Windows Server 2008 which is required to mount VHDs on a Windows Server 2008 machine while it remains off for all other performance testing scenarios.

Choice of Workloads

We used two types of performanceworkloads. The advantage of these workloads is that they are neutral to operating systems and underlying storage platforms. This makes it easier to compare VHD with other virtual disk formats.

- Monolithic I/O-Based Workloads

The workloads here consist of single type of I/O. We have been using these monolithic types of I/O workloads since they are relatively simple and quick to configure and run, and give a good baseline figure to easily identify the impact of changes during the development cycle.

Sequential Reads & Writes	4K	64K	256K	1M
Random Reads & Writes	4K	8K	32K	64K

The two overlapping I/O sizes (4K/64K) here also give us another opportunity to compare the performance of sequential I/O with random ones.

- Application-Based Workloads

We picked some of most commonly used server workloads to simulate enterprise customer scenarios. They are well-recognized by the storage community and adopted by the industry.

Workload Category	I/O Size	Percentage of READ vs. WRITE	Percentage of RANDOM vs. SEQUENTIAL
Web File Server	4KB	95% RD vs. 5% WR	75% RAND vs. 25% SEQ
Web File Server	8KB	95% RD vs. 5% WR	75% RAND vs. 25% SEQ
Web File Server	64KB	95% RD vs. 5% WR	75% RAND vs. 25% SEQ
Decision Support System DB	1MB	READ	RANDOM
Media Streaming	64KB	98% RD vs. 2% WR	SEQUENTIAL
SQL Server Log	64KB	WRITE	SEQUENTIAL
OS Paging	64KB	90% RD vs. 10% WR	SEQUENTIAL
Web Server Log	8KB	WRITE	SEQUENTIAL
OLTP DB	8KB	70% RD vs. 30% WR	RANDOM
Exchange Server	4KB	67% RD vs. 33% WR	RANDOM
Workstation	8KB	80% RD vs. 20% WR	80% RAND vs. 20% SEQ
Video on Demand	512KB	READ	RANDOM

- I/O Queue Depth: 1 to 256

Measuring I/O performance with various queue depths is another key to scrutinizinghow native VHD performs with underlying platforms, e.g. how quickly we can hit the saturation point. It can also help determine how efficiently the native VHD driver stack handles outstanding I/Os as the queue depth increases. In addition, it can help us determine if there are any hidden performance issueswhena certain pattern is not observed in the results. For example, before the target is saturated, generally we should see the throughput increase as more outstanding I/Os are issued.

Choice of Performance Metrics

Twomajor performance metrics are considered in this report: throughput and latency.

Throughput is measured either in terms of I/Os per second (IOPS) or megabyte per second (MBPs). Typically undera monolithicI/O workload, for small sized I/Os, we intend to use IOPS while reservingMBPS for larger sizedI/Os. For Application based workloads, we use IOPS exclusively for performance comparisons. Higher throughput means better performance.

Latency (average I/O response time) is measured in terms of milliseconds. Combined with throughput, we can use latency to easily identify the storage saturation point when throughput gets flatand latency keeps going up.Lower latency means shorter response time and better performance.

Choice of Target File Size

Since we're comparing raw disk vs. raw file vs. VHD, it becomes necessary to set preferred size to be equal or very close to each other to make such type of comparison meaningful.

- Random performance is largely impacted by the target size.
- Minimizing the performance impact of the on-disk layout for files is another concern (i.e. there can be a performancegap between an outer track of a physical disk drive and an inner track). Creating a large file to occupy the entire diskguarantees a consistent disk layout.
- Minimizing the impact of hardware caches(depending on the RAM size) is also important so the entire file cannot be easily fit or pre-fetched into the cache to get false performance results.

Choice of Data Block Size

The default data block size for dynamically expanding and differencing VHDschanged from 512KB in Windows Server 2008 to 2MB in Windows Server 2008 R2. We made the change in default block size based on extensive testing across numerous scenarios. Based on our experimental results we found 2MB to be a better default. This does not mean it is better for all scenarios and in particular there are some simple primitive tests where 512KB is a better choice. However the primitive tests are generally not representative of what real workloads do.

This leads to the question when comparing Windows Server 2008 and Windows Server 2008 R2 if we should use the default setting or use the same block size in all tests. We chose to use the default because this is the finely tuned parameter and we want to compare the out-of-box performance because this is what most users will experience.

Choice of VHD AttachMode

The goal of this report is to focus on VHD driver stack performance and treat VHDsjust like regular disk drives. Measuring VHD performance directly on parent partition has the following benefits:

- Only VHD related components are involved. No Hyper-V storage stack is needed.
- The performance results are comparable to a raw file sitting on top of physical file system.
- The performance impact of a guestoperating system is eliminated.

Unless indicated otherwise, the VHDbeing testedis attached as a loopback drive on the parent partition, and then the loopback drive is used as a raw disk during the performance measurement process (just like how we measure the performance of a physical drive). We use virtual machine mode when there is a need to compare performance between the parent partition and avirtual machine.

Reducing variation in measurements

In reality, an enterprise server may be engaged inmany computingactivities simultaneously whendealing withthe disk I/O subsystem. To minimize any interference from other applications and get accurate performance results, we take explicit action to reduce the overhead of other applications and services.For example, if we leave Hyper-V Manager open during I/O performance measurement within VM, Hyper-V Manager will periodically communicate with the VM to get the screen contents update. This can produce skewed results varying the accuracy by as much as 5 to 10%.

B. Test tools

IOMeter is a popular I/O benchmarking tool in storage community and it is also thetool used to measure native VHD performance. Here are the major IOMeter settings being used in this report:

- Version: 2008-06-22RC2(x64)
- Ramp-up Time: 30 sec
- Test Run Time: 60 sec
- Number of IOMeter Worker: 1
- I/O Alignment Size: Sector(512B)

C. Server Platformand Operating System

We used an Intel Nehalem-EP server with dual quad-core (2128MHZ) processors, 6GB RAM with NUMA enabled and a 1TB SATA RAID0 based system drive. Windows Server Enterprise 2008(x64)and Windows Server Enterprise 2008 R2 (x64)are operating systems were used for all performance experiments.

D. Storage Platform Settings

To simplify the performance measurement environment, we used Direct–Attached Storage (DAS)from Dell as our target. DAS is not as common as SAN under enterprise scenarios. However, DAS typically has fewer performance factors (e.g. block size, thin provisioning, de-duplication etc.) to get involved so we can purely focus on software stack performance and eliminate platform dependent factors as far as possible. The performance results collected onDAS are generally more sustainable due to the similar reasons. Certain features existing in SAN environment may impact actual VHD performance results. For example, you may see performance penalties caused by misalignment of dynamic VHDson certain type of SANswhich may not show up on a DAS platform

Here are the general settings:

Storage Platform Hardware Settings

- Enclosure: Dell PowerVault MD1000
  - Capable of holdingup to 15 3.5” drives without expansion
  - Single pathwithout built-in RAID controller cards

- Hard Disk Drives: Seagate Cheetah SAS Drive (Quantity:15)
  - Model: ST3146356SS
  - 15000 RPM, 3.5”, 146GB, 16M Disk Cache
  - Sustained data transfer rate for single drive is approximately 160MBps.

- Native physical sector size: 512Bytes

- RAID Controller Card: LSI 8880EM2
  - LSI1078 Storage Processor, PowerPC 500MHZ
  - Supports 3Gbps SAS/SATA drives
  - Supports eight external SAS/SATA ports
  - 512MB DDR2 Cache (667MHZ),
  - PCIe x8
  - Based on performance measurements, the maximum bandwidth the LSI 8880EM2 (mainly determined by storage processor) can sustain is approximately 1600MBps for reads 700MBps for writes.

Storage Platform Software Settings

- Seagate SAS drive firmware: HS0C

This was the latest firmware available for these drives at the time of publication.

- RAID Controller firmware: Mega RAID(3.6) 11.0.1-0013

This was the latest firmware published by LSI wecould use to flash this RAID controller card and keep it up-to-date.

- LSI Mega SAS Driver

Making sure both Windows Server 2008R2 and Windows Server 2008 use the same third party mini-port driver is crucial for our performance comparison purpose.

  - Windows Server 2008 R2 ships with the latest driver version available inbox at the time of writing, 4.5.0
  - Windows Server 2008 ships with version 2.3.0. We were able to upgrade to 4.4.0and there is no performance difference between 4.4.0 and 4.5.0.

- RAID Array Settings

We used a 15-spindle RAID0 configuration. RAID array properties are detailed below. A 256K stripe unit size was chosen based on internal performance investigations for LSI RAID controller cards. We're attempting to bypass cache effects along the I/O path by using write-through for disk controller cache and disabling disk drive cache.

  - RAID Level: RAID 0
  - Size:2190GB (146GB x 15)
  - Stripe Size: 256 KB per disk
  - Number Of Drives:15
  - Cache Policy: WriteThrough, ReadAhead, Direct I/O
  - Disk Cache Policy: Disabled

Note: enabling disk cache does bring up additional performance benefits as long as file system flush and disk controller flush frequency are well set. In production environment, we do recommend customers to enable controller cache when a battery backup unit is available.

We chose RAID0 for two reasons:

- RAID0 gives the best performance amongst all RAID levels
- No parity calculation is required, and thus no additional I/Os are generated for each write. This can help simplify performance overhead calculations.

E. VHDDetails

The introduction gave an overview description of the different types of VHD. This section givesmore detail on the format and impact on overall performance and usage. The full VHD specification is available online at http://technet.microsoft.com/en-us/virtualserver/bb676673.aspx .

- Fixed sized VHDs

The file structure of fixed sized VHDs is relatively straightforward. Fixed sized VHDs do not contain any data blocks and they are essentially a flat file plusfooter. We recommend using fixed size VHD when performance is the top concern for customers. From an implementation perspective, the VHD driver itself does not impose any file size limitation (except the one from NTFS, i.e. 16TB). In this report, we created a fixed VHD which is about 2040GB, the maximum size supported by Hyper-VManager UI.

- Dynamically expanding VHDs

Dynamically expanding VHDs can grow dynamically without requiring space allocation up-front. The following table shows the actual physical space occupied by a fully populated 2040GB dynamically expanding VHD. In Windows Server 2008 R2, we changed the default block size from 512KB to 2MB for performance reasons. However, the block size change also has animpact on the actual physical file size. Smaller blocks size means more data blocks are needed for a pre-defined virtual capacity. Thus more space has to be reserved for the Block Allocation Table (4 bytes per entry) and Sector Bit Map (512B regardless of block size for alignment purpose). For example, a 2040GB dynamically expanding VHD in Windows Server 2008 has to reserve 16MB for BAT and 2GB for sector bit maps. From the dynamically expanding VHD perspective, these sector bit maps are really not as useful as they are in a differencing VHD and could be optimized.

	2040GB Dynamically Expanding VHD in Windows Server 2008 R2	2040GB Dynamically Expanding VHD in Windows Server 2008
VHD Capacity	2040GB	2040GB
Data Blocks within VHD	1044480	4177920
BAT (Block Allocation Table)	Approximately 4M(default: 2M/Block)	Approximately 16M(default: 512K/Block)
Sector Bit Maps (512B/Block)	Approximately 510M	Approximately2G
Footer/Header	2K	2K

- Differencing VHD

A Differencing VHD is very much like the dynamic VHD as they are both dynamically expandable. Their internal structures remain largely the same although runtime interpretation (e.g. sector bitmap) may be different. You can convert a differencing VHD to a dynamic VHD by choosing to merge to a new file. But there are also notable differences between them. For example, you can expand a dynamic VHD (in terms of virtual size) but that is not allowed for differencing VHDsdue to the inherent size limitation inherited from the parent. Adifferencing VHD also requires the presence of its parent to perform I/O activity.

Differencing VHDsarethe foundation for Hyper-V VM snapshots and many other customer scenarios (e.g. multiple client images share the same golden parent image). In the previous section, we laid out details about physical space usage for fully populated dynamically expanding VHDs. The Differencing VHD format has added a little bit more overhead in terms of actual physical file sizebecause it has to use built-in parent locators to save the parent path information. The absolute parent locator path within a differencing VHD is set to 512B by defaultand the relative parent locator path is set to 64KB(this one is indeed optional if there the differencing VHD does not share any common path with the parent [for example when they are located on different volumes]). When the native VHD stack tries to open a differencing VHD, both the differencing and the parent VHD are opened. The parent locator information may get updated when the parent location information is changed. For example, in case the differencing chain is broken, areconnect operation lets you specify the location of parent and re-establish the link between parent and child.

When accessing differencing VHDs, parent VHDsareopened in read only mode by the driver and the child VHDs are opened in read/write mode so all the changes will stay within the child. However, the VHD format and VHD driver assume that parent VHD will not change and have no inherent protection against that assumption. In other words, it is possible to modify a parent VHD which will corrupt the relationship between it and any child VHDs. A differencing VHD merge into a parent also corrupts all other sibling VHDs not on the merging path regardless of merging success or not. The child and the other siblings may never be made aware of if there are changes made in their parent and results may become unpredictable.

A differencing chain can consist ofone or more VHDs. However, the VHD specification and native VHD driver does not requires the data block size for the entire chain be the same.For best practices, we recommend using 2MB for both the parent and child VHDs to improve performance as the reads to the parent may be impacted due to a smaller block size. For certain scenarios like booting a differencingVHD on a physical machine, the child and parent VHDsmust use the same data block size (note that this rule does not apply to booting a Virtual Machine in Hyper-V).In Windows Server 2008 R2, differencing VHDs created via DiskPartor Hyper-V Manager always uses the same data block size as the parent.That means you will get a 512KB differencing VHD instead of the default 2MB even in Windows Server 2008 R2 if that is the data block size of the parent VHD. If parent is a fixed sized VHD, then the default data block size will be always used for the differencing VHD.

During performance experiments, the differencing chain length is set to one, i.e. one parent and one differencing VHD (child). A2040G dynamically expanding VHD is chosen as the parent and remains unpopulated. We then fully populate the differencing VHD to make it as big as possible on the physical volume.

F. File System and Volume Settings

VHD only works with NTFS and is generally considered as a large file in terms of average file size. In this report, wealways chose 64K as the NTFS cluster (Allocation Unit) size for VHD and Non-VHD file performance measurement. That also aligns with SQL Server performance best practices when a large database file is used.All the physical volumes remain clean: a fresh format is performed before creating VHD files to minimize the performance impact of fragmentation.

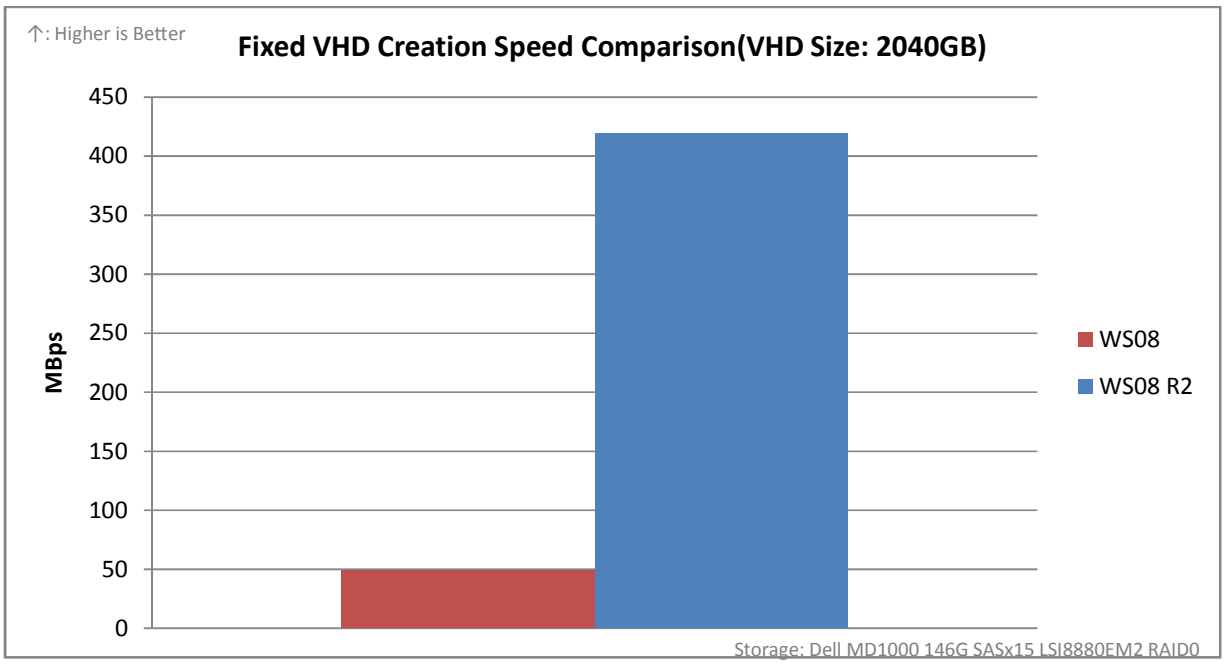
3. Comparison of Windows Server 2008 andWindows Server 2008 R2 VHD Performance

The following section compares Windows Server 2008 and Windows Server 2008 R2 overallVHD performance. We look at creation speed, various workloads, impact of queue depth, block size, chain depth and other parameters that impact overall VHD performance.

A. Fixed Size VHD creation speed

Creating new VHDs generally happens when avirtual machine is provisioned or a new boot environment is needed. The time it takes to create dynamically expandingVHDs and differencing VHDs is on the order of a couple of seconds. However,fixed sized VHDs require the entire size of the virtual disk to be written during creation. This means the fixedsized VHD creation speed is determined by the time to zero out all the space allocated upfront.

Below is a 2040GB fixed VHD creation speed comparison based on a directly-attached storage between Windows Server 2008 and Windows Server 2008 R2. Note: the speed improvement may vary on different storage platforms and VHD file size.

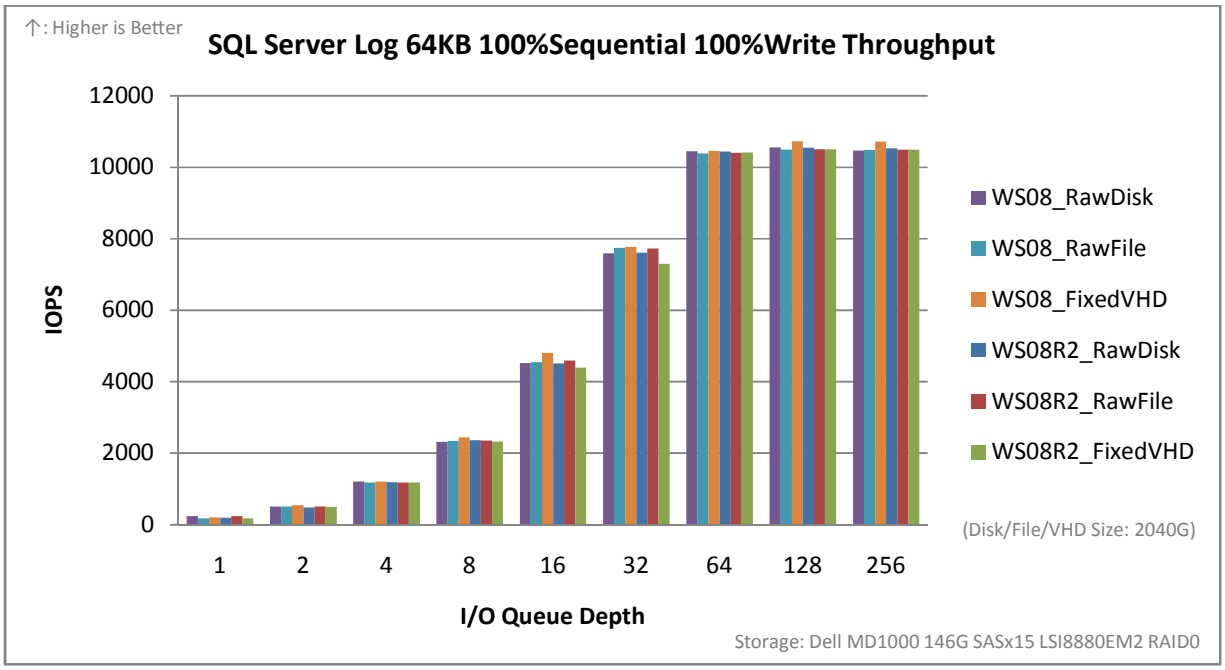


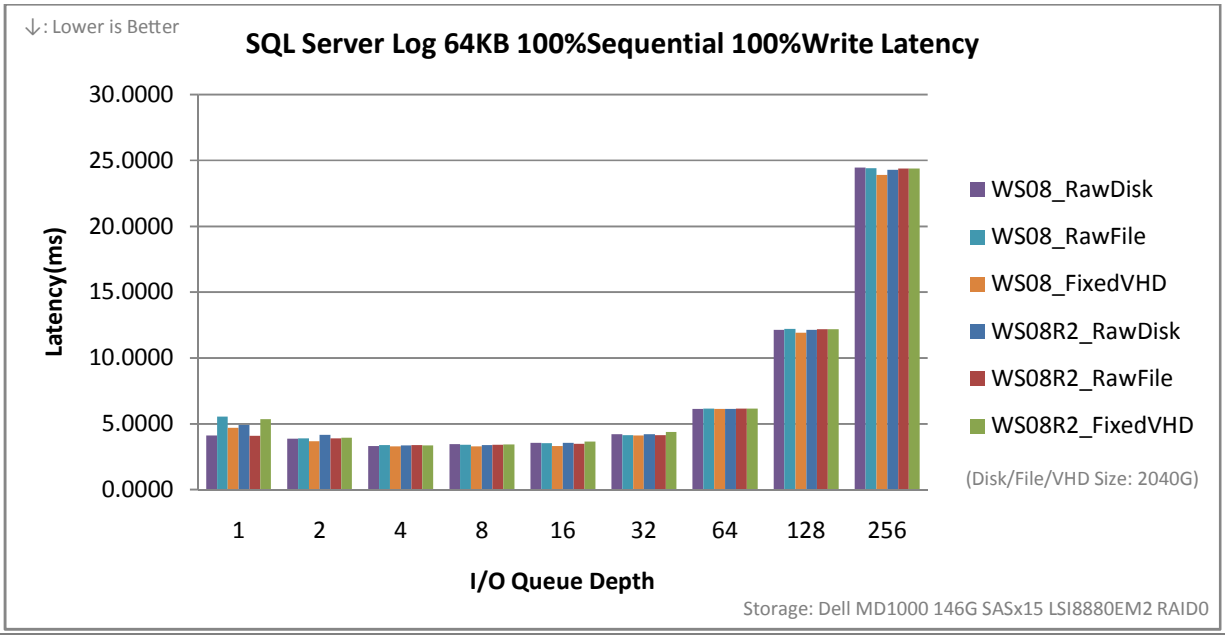
B. Fixed Sized VHD Performance Comparison

The fixed sized VHD performance has been on-par with the physical disk since Windows Server 2008/Hyper-V release to manufacturing InWindows Server R2 fixed VHD performance remains intact, i.e. it as good as raw disk or raw file.

SQL Server Log Workload

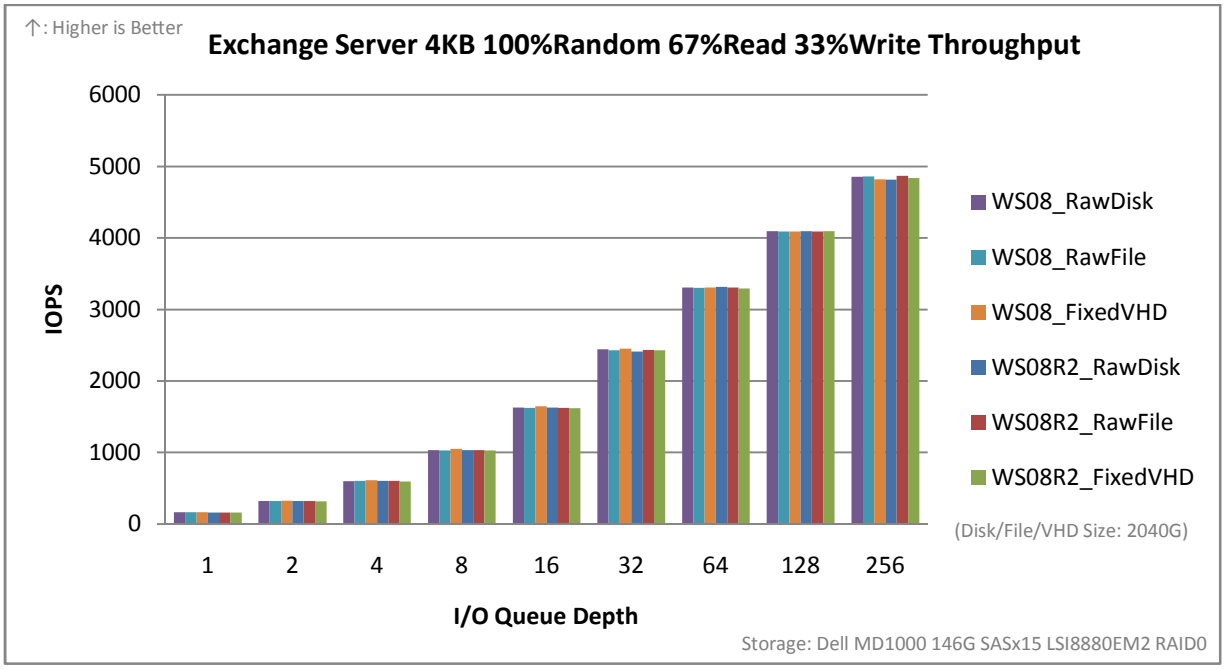
- **64KB** I/O size, **0%** Read, **100%** Write, **0%** Random, **100%** Sequential

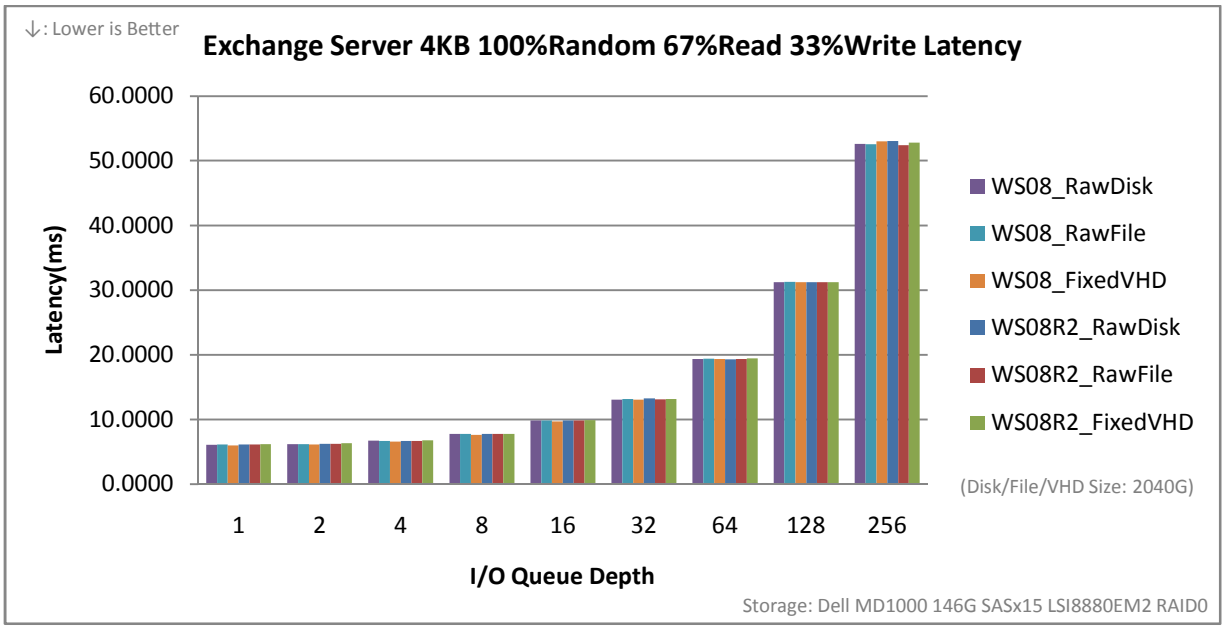




Exchange Server Workload

- **4KB** I/O size, 67**%** Read, 33**%** Write, **100%** Random





C. Dynamically ExpandingVHD PerformanceComparison

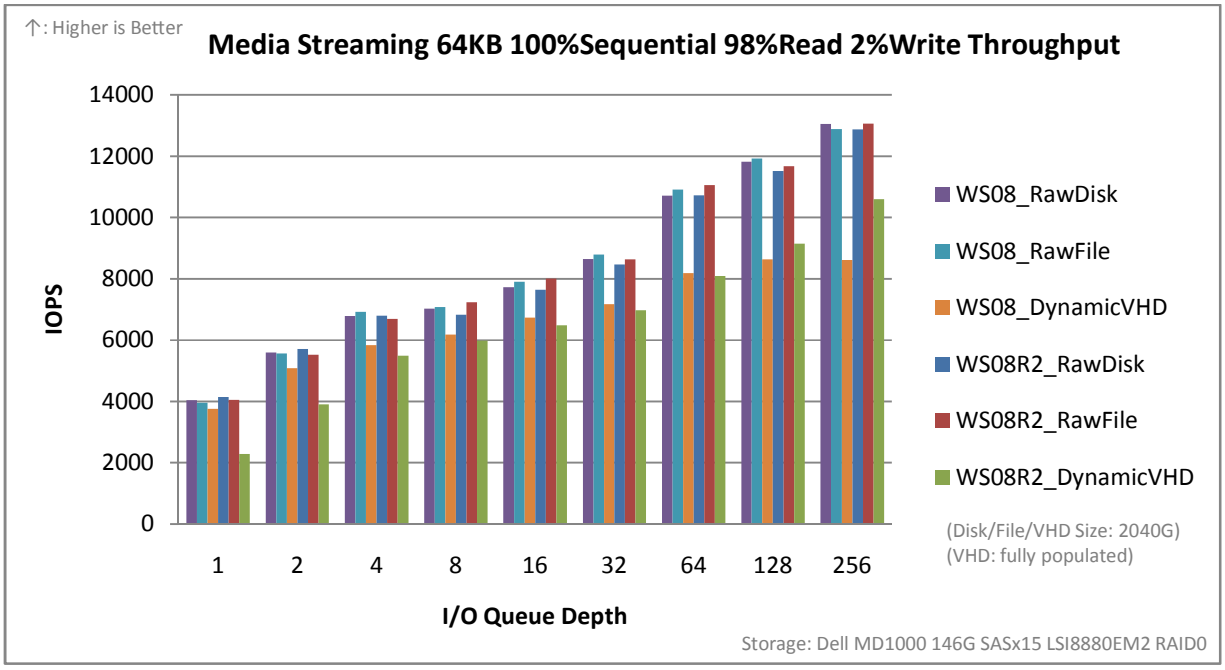
The following table shows a summary of*peak*IOPS comparison for a fully populated 2040GB dynamically expanding VHD using a Dell MD1000 DAS. Note: the queue depth corresponding to the peak IOPS may differ when comparing Windows Server 2008 R2 with Windows Server 2008.

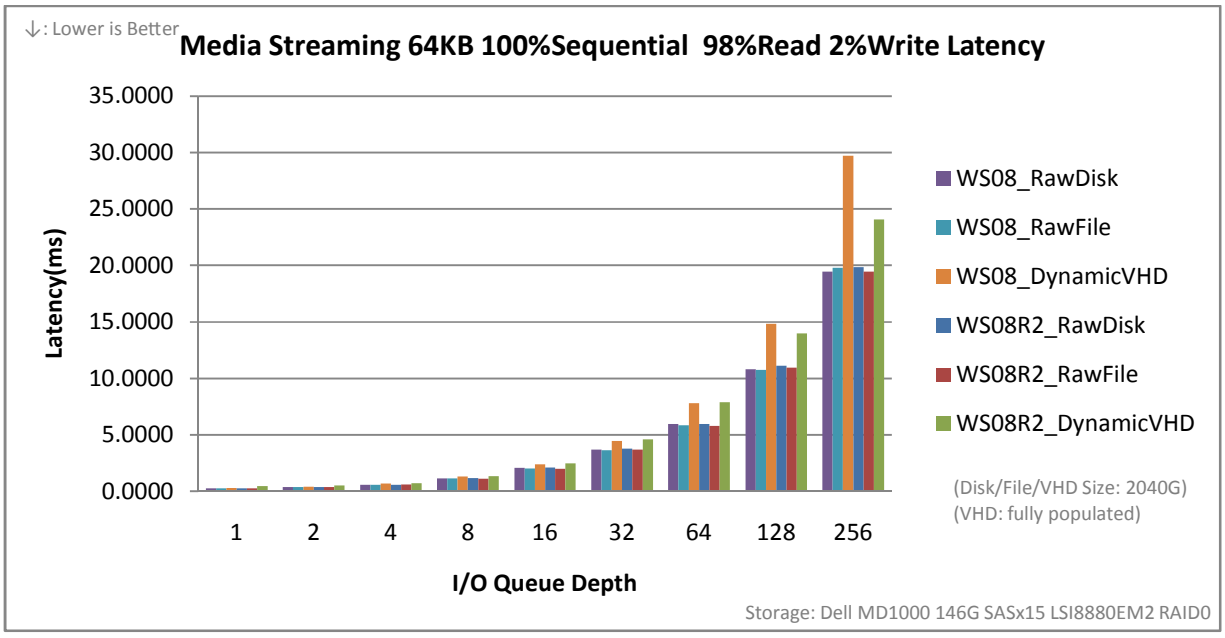
Peak IOPS	Windows Server 2008	Windows Server 2008 R2
4K Sequential Reads	52070	92067
4K Sequential Writes	7528	43892
64K Sequential Reads	24599	24631
64K Sequential Writes	1083	10302
256K Sequential Reads	6262	6288
256K Sequential Writes	371	2500
1M Sequential Reads	1565	1563
1M Sequential Writes	85	616

Peak IOPS	Windows Server 2008	Windows Server 2008 R2
4K Random Reads	1667	5165
4K Random Writes	157	2611
8K Random Reads	1734	5064
8KRandom Writes	155	3117
32K Random Reads	1642	4405
32K Random Writes	146	3079
64K RandomReads	1506	3761
64K Random Writes	137	2888

Media Streaming Workload

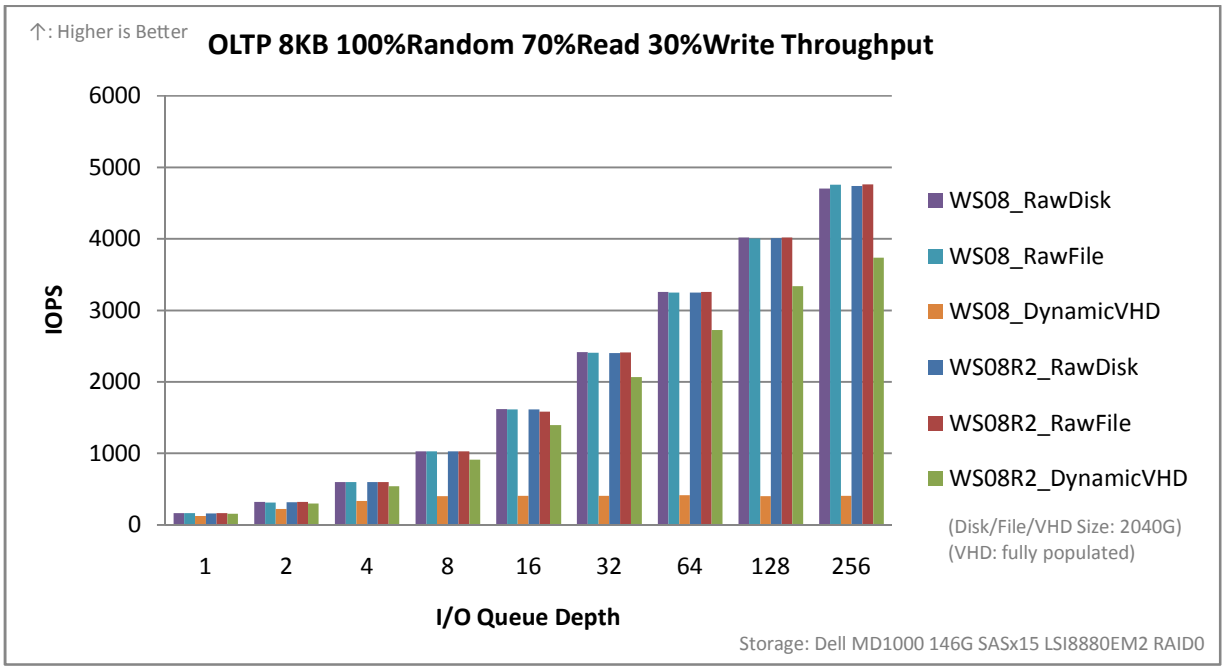
- **64KB** I/O size, **98%** Read, **2%** Write, **0%** Random, **100%** Sequential

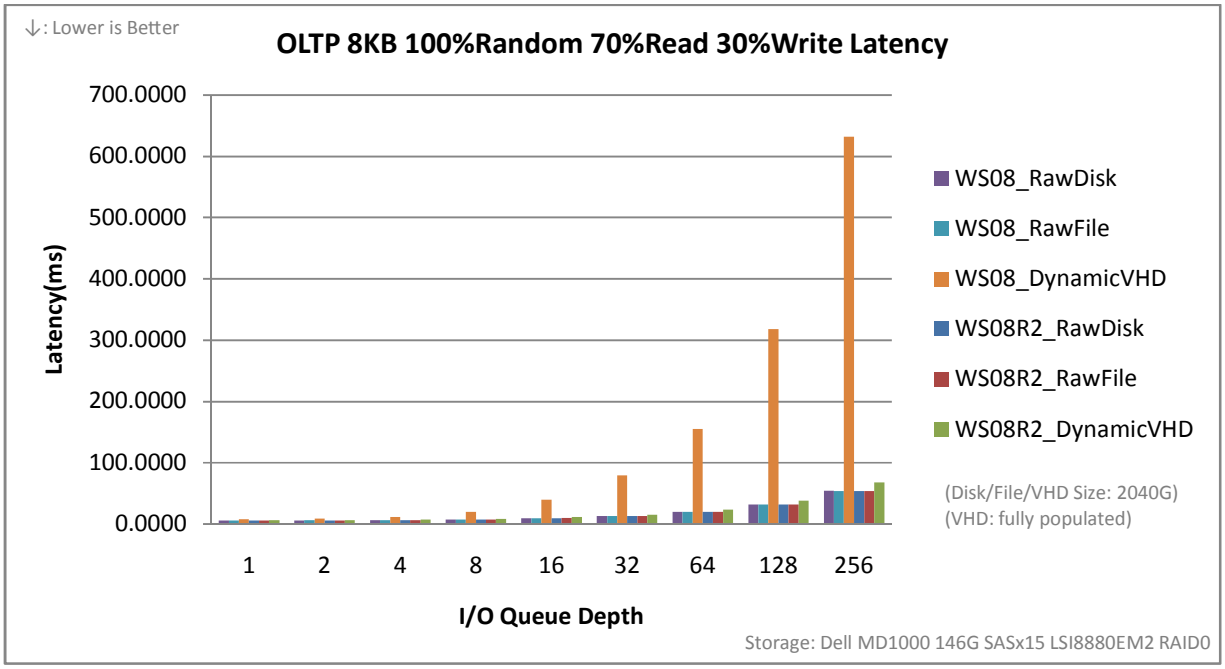




Online Transaction Processing Workload

- **8KB** I/O size, **70%** Read, 3**0%** Write, **100%** Random, **0%** Sequential





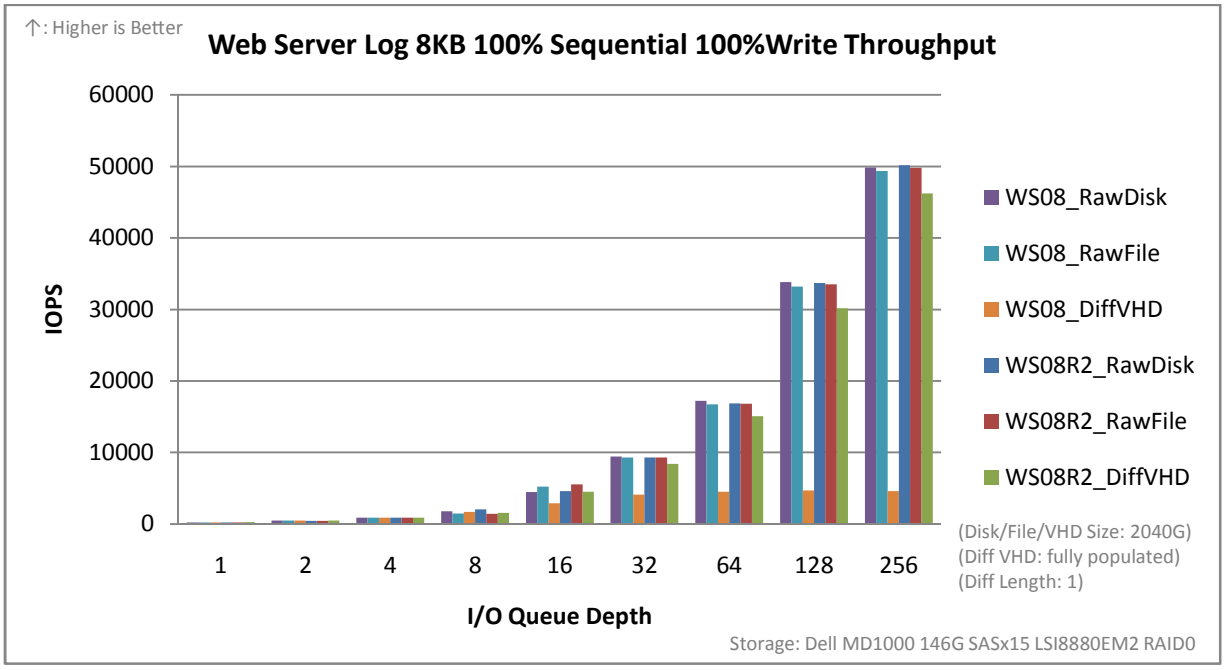
D. Differencing VHD Performance Comparison

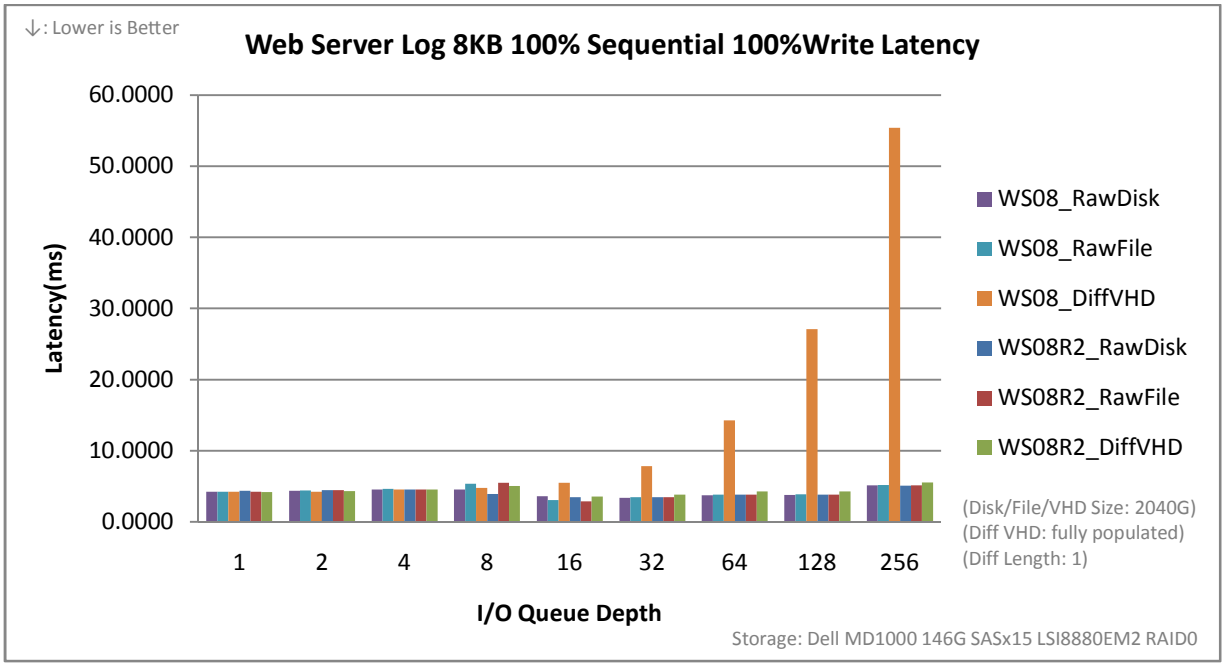
The following table shows a summary of*peak*IOPS comparison for a fully populated 2040G differencing VHD (depth: 1) using a Dell MD1000 DAS. Note: the queue depth corresponding to the peak IOPS may differ when comparing Windows Server 2008 with Windows Server 2008 R2.

Peak IOPS	Windows Server 2008	Windows Server 2008 R2	Peak IOPS	Windows Server 2008	Windows Server 2008 R2
4K Sequential Reads	39908	93352	4K Random Reads	159	5167
4K Sequential Writes	7320	44933	4K Random Writes	156	2673
64K Sequential Reads	6782	24651	8K Random Reads	157	5033
64K Sequential Writes	1076	10321	8KRandom Writes	155	3129
256K Sequential Reads	1414	6294	32K Random Reads	150	4410
256K Sequential Writes	364	2511	32K Random Writes	146	3077
1M Sequential Reads	355	1565	64K RandomReads	142	3760
1M Sequential Writes	84	629	64K Random Writes	136	2904

Web Server Log Workload

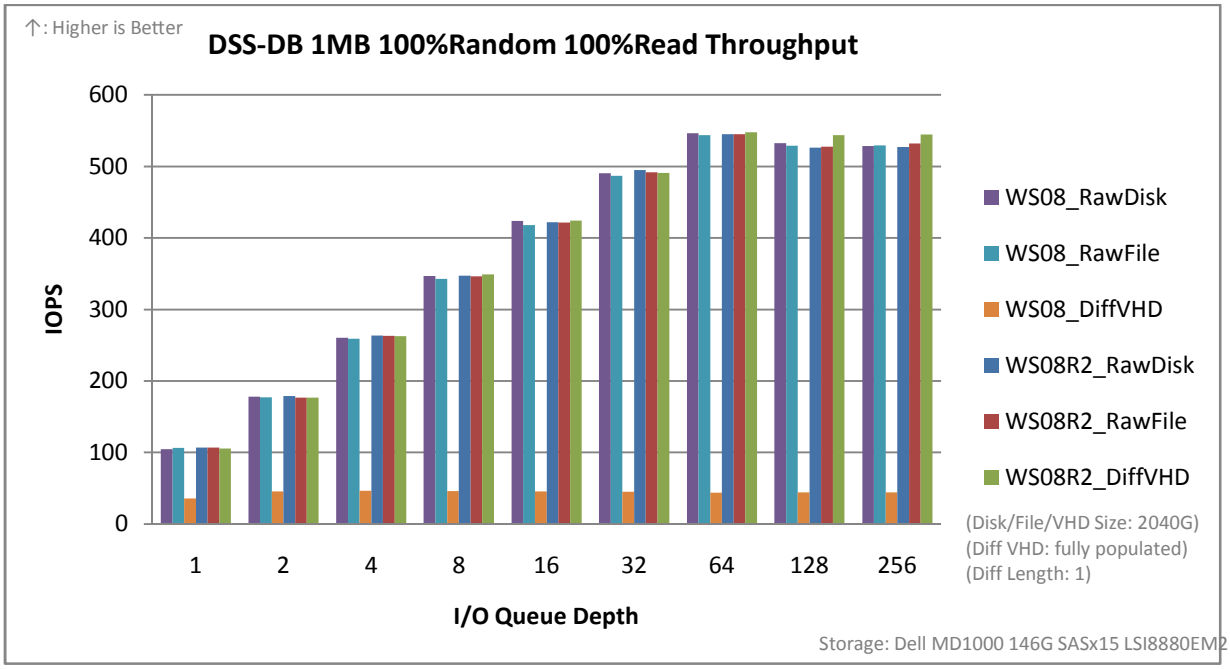
- **8KB** I/O size, **0%** Read, **100%** Write, **0%** Random, **100%** Sequential

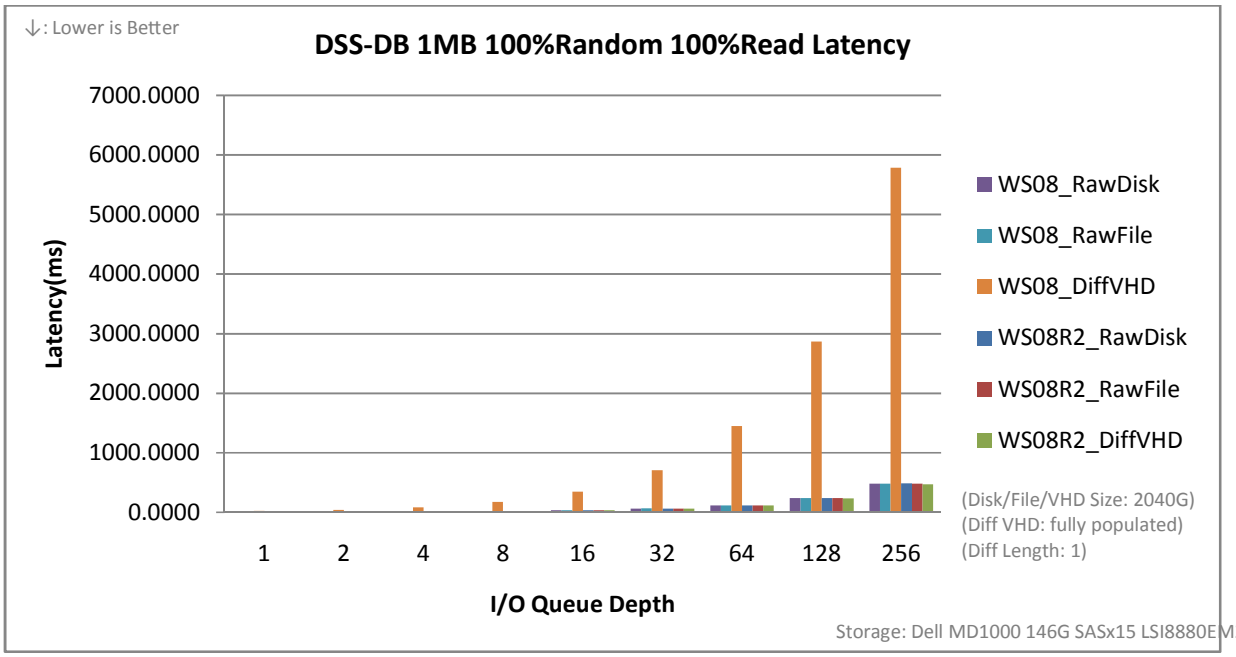




Decision Support System Database Workload

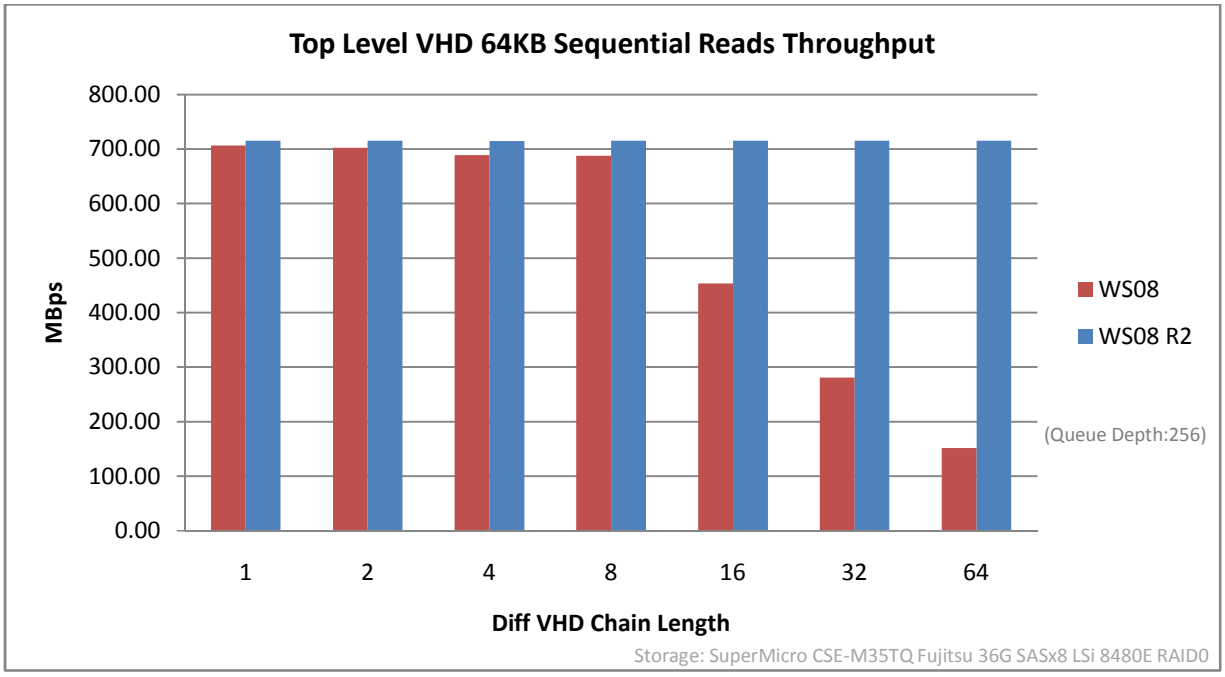
- **1MB** I/O size, **100%** Read, **0%** Write, **100%** Random, **0%** Sequential

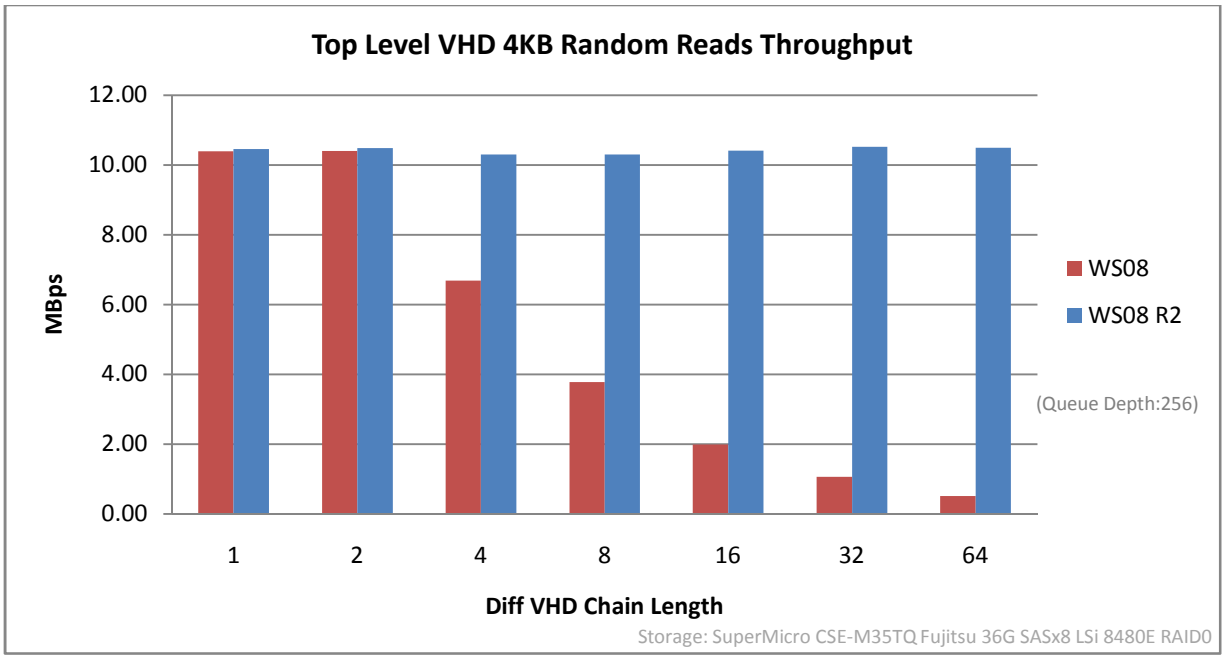




Performance Impact of Differencing Chain Length

In Windows Server 2008, experiments uncovered a critical performance issuein differencing VHD: performance of accessing VHDs that belong to a chain of differencing disks degraded very quickly as differencing chain length increased. This issue has been addressed in Windows Server 2008 R2. The following performance results show the read performance improvements (all writes are directed only to the last VHD in the chain) to the top level VHD as the differencing chain length increases.



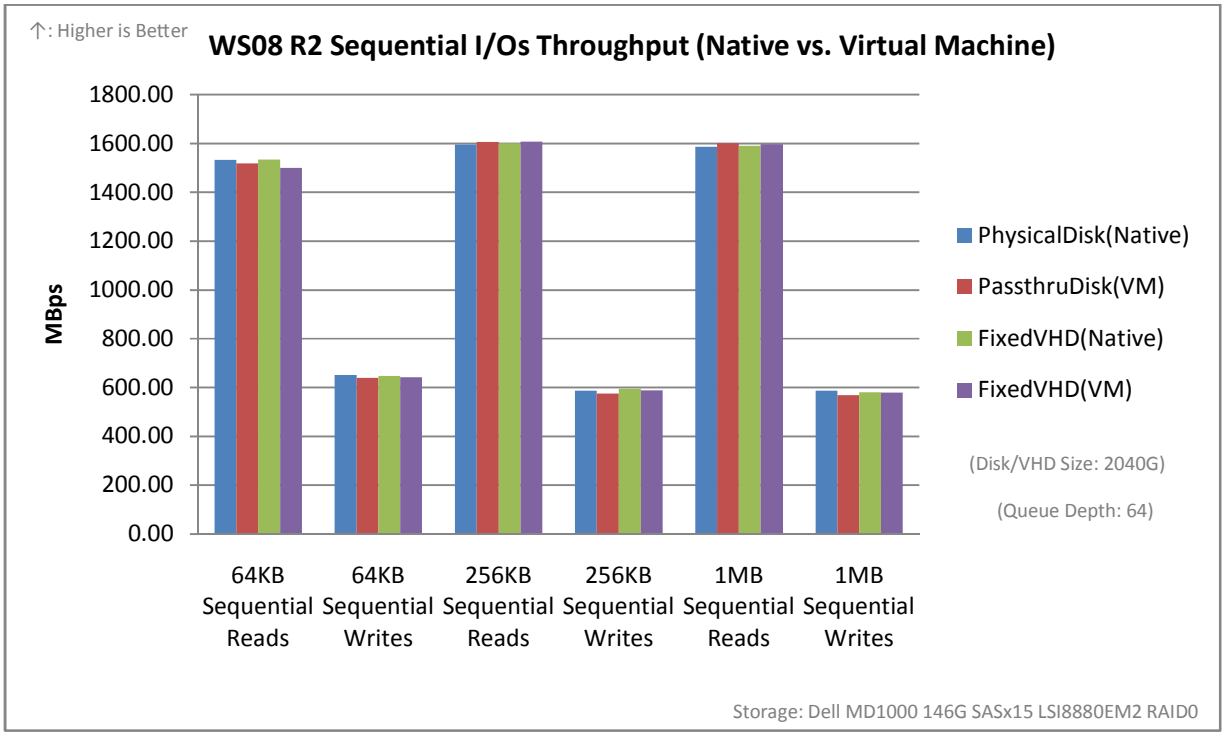


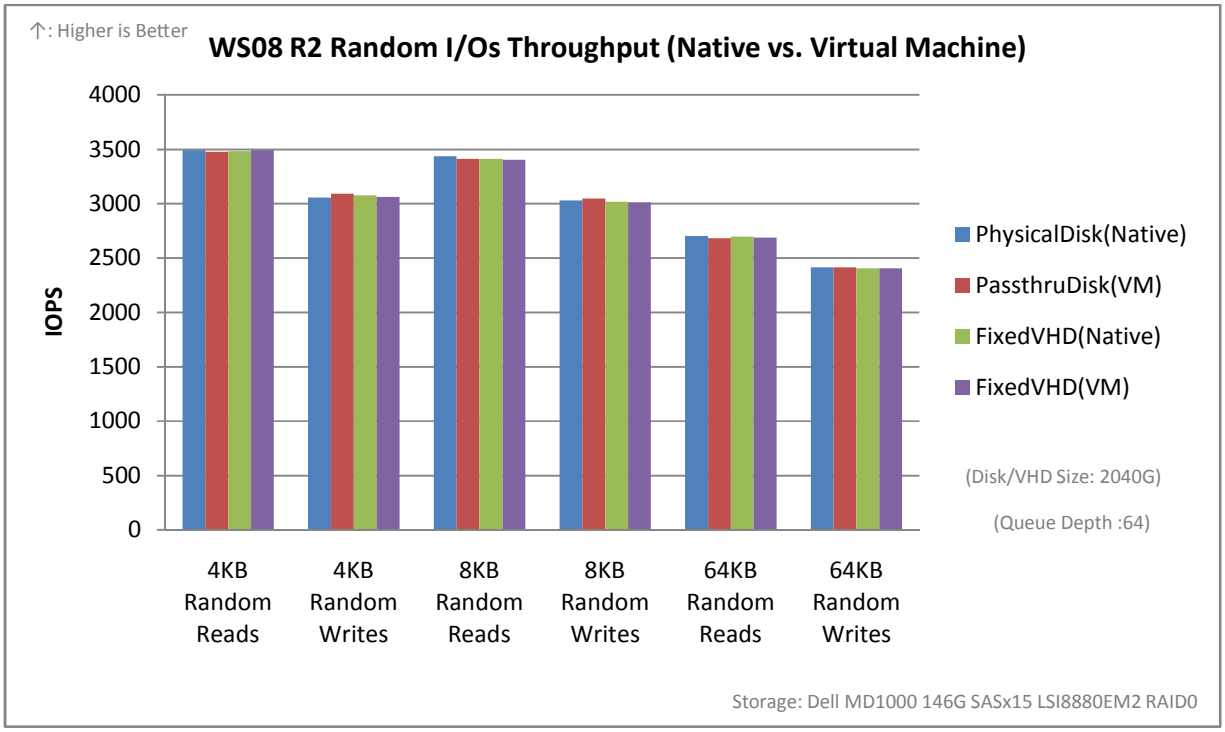
4. Comparison of Native and Virtual Machine VHD Performance

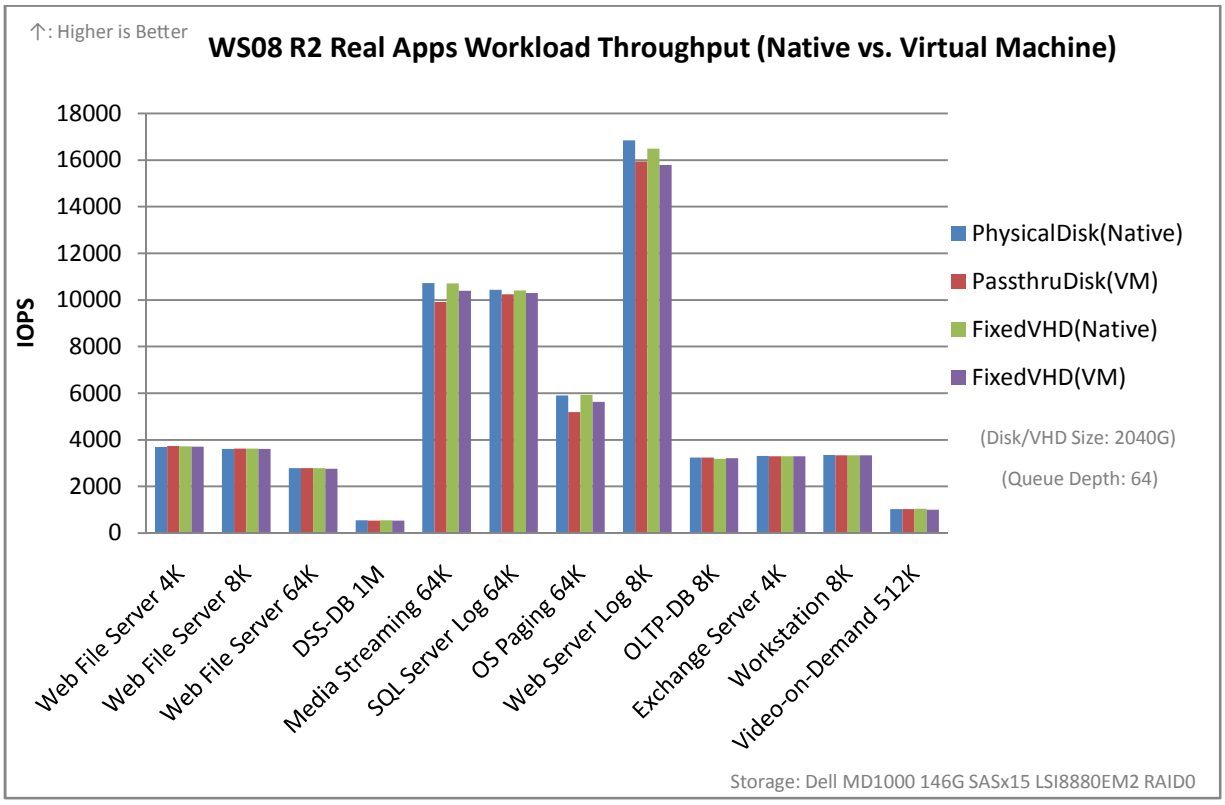
We have been focusing on the performance comparison between physical disk, raw file and a native VHD all residing on the parent partition. However, VHD was originally designed and developed as the storage container format for virtualization environments. Virtual Machines remain the most common use for VHD deployment.In this section, we compare the I/O performance between native storage (physical disk and native VHD) and virtualized storage (pass-through disk and Virtual Machine VHDs).

E. Hyper-V Virtual Machine Performance Testing Settings

- Guest Operating System: Windows Server Enterprise 2008 R2 x64
- RAM: 1024MB
- Number of Virtual Processors: 1
- Guest Storage Controller
    - IDE: system drive for booting guest OS
    - SCSI: data drive for performance measurement


↑: Higher is Better
WS08 R2 Sequential I/Os Throughput (Native vs. Virtual Machine)
MBps
1800.00
1600.00
1400.00
1200.00
1000.00
800.00
600.00
400.00
200.00
0.00
64KB Sequential Reads
64KB Sequential Writes
256KB Sequential Reads
256KB Sequential Writes
1MB Sequential Reads
1MB Sequential Writes
PhysicalDisk(Native)
PassthruDisk(VM)
FixedVHD(Native)
FixedVHD(VM)
(Disk/VHD Size: 2040G)
(Queue Depth: 64)
Storage: Dell MD1000 146G SASx15 LSI8880EM2 RAID0


↑: Higher is Better
WS08 R2 Random I/Os Throughput (Native vs. Virtual Machine)
IOPS
4000
3500
3000
2500
2000
1500
1000
500
0
4KB Random Reads
4KB Random Writes
8KB Random Reads
8KB Random Writes
64KB Random Reads
64KB Random Writes
PhysicalDisk(Native)
PassthruDisk(VM)
FixedVHD(Native)
FixedVHD(VM)
(Disk/VHD Size: 2040G)
(Queue Depth :64)
Storage: Dell MD1000 146G SASx15 LSI8880EM2 RAID0

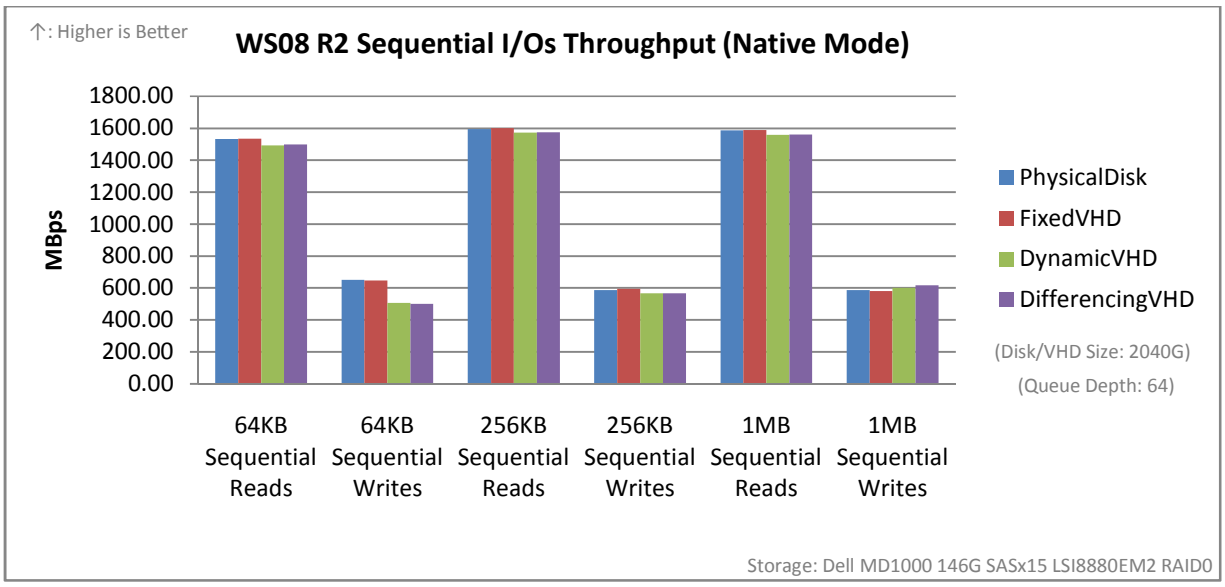


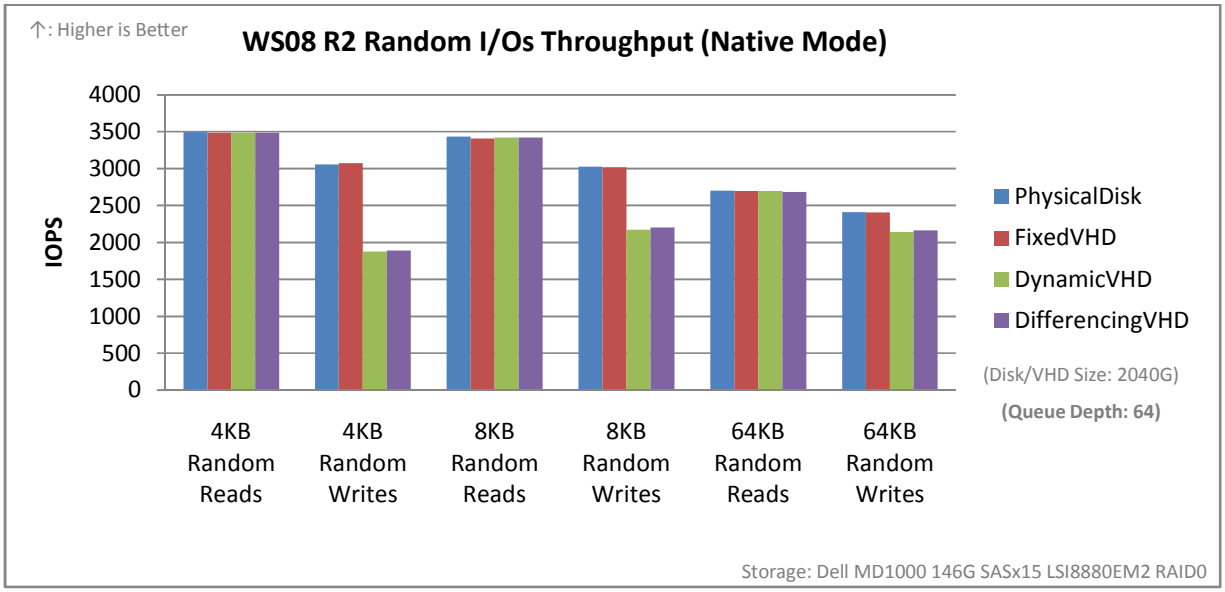
The graphshere showthe performance gap between a physical disk, pass-thrudisk and a fixed VHD (either attached to native or a VM) is small. The potential performance difference not shown here, may come from the path length (i.e. CPU cycles per I/O) as a pass-throughdisk generally requires less CPU resources due to the shorter path compared with fixed sized VHD.

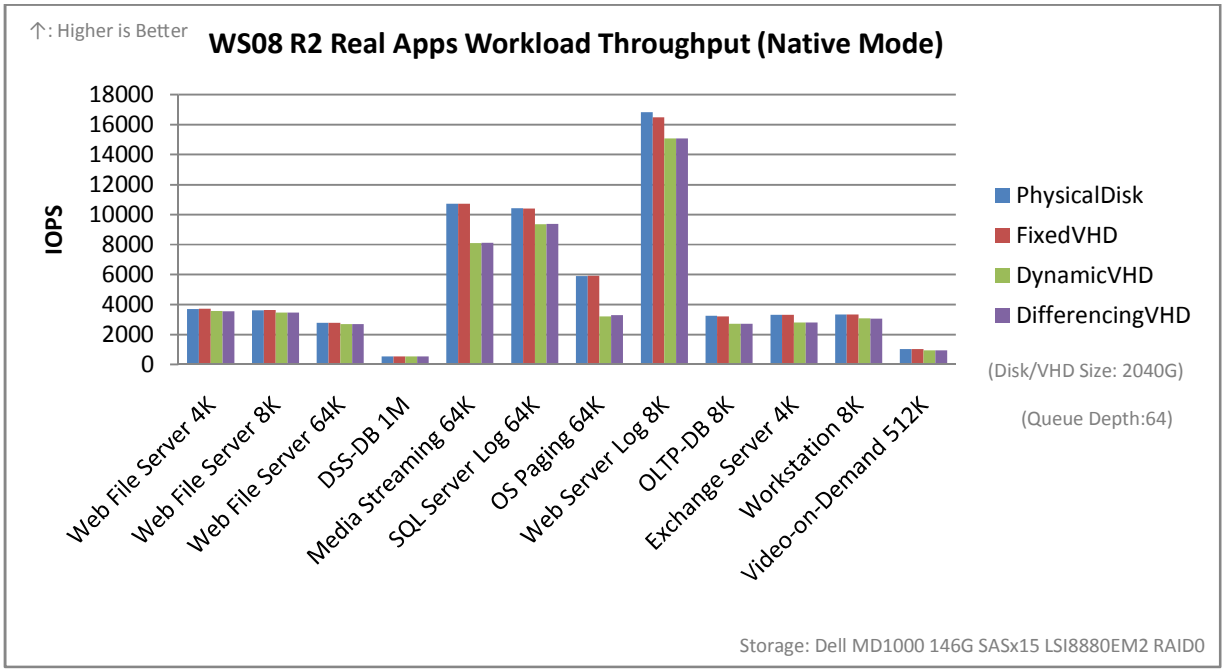
5. Comparison of VHD type performance [fixedsized vs. dynamically expanding vs. differencing]

DynamicallyexpandingVHDs are not recommended for virtual machines that run server workloads in a production environment. Fixed VHD is preferredconsidering it has better performance with committed space allocation so it will not run out of physical backing storage.

The following graphs show, in Windows Server 2008 R2, the gap between a fixed VHD and a fully populated dynamic or differencing VHD is fairly small. The major performance gap comesfrom small I/O writes.However, that gap may be widened if the VHD has to expand itself first before taking any user data.






↑: Higher is Better
WS08 R2 Real Apps Workload Throughput (Native Mode)
IOPS
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Web File Server 4K
Web File Server 8K
Web File Server 64K
DSS-DB 1M
Media Streaming 64K
SQL Server Log 64K
OS Paging 64K
Web Server Log 8K
OLTP-DB 8K
Exchange Server 4K
Workstation 8K
Video-on-Demand 512K
PhysicalDisk
FixedVHD
DynamicVHD
DifferencingVHD
(Disk/VHD Size: 2040G)
(Queue Depth:64)
Storage: Dell MD1000 146G SASx15 LSI8880EM2 RAID0

6. How to choose your Hyper-Vand VHD Storage Container Format

Customers always have to make a choice when they need to decide what the appropriate storage container format is for deploying virtual machines using Hyper-V. The following summary table is intended to make the decision-making process easier:

Storage Container	Pros	Cons
Pass-through Disk	• Fastest performance • Simplest storage path becausefile systemon host is not involved. • Better alignment under SAN. • For shared storage based pass-through, no need to mount the file system on host and that may speed up VM live migration. • Lower CPU utilization • Support very large disks	• VM snapshot cannot be taken • Disk is being used exclusively and directly by asingle virtualmachine. • Pass-through disks cannot be backed up by the Hyper-V VSS writer and any backup program that uses the Hyper-V VSS writer.
Fixed sizedVHD	• Highest performance of all VHD types. • Simplest VHD file format to give the best I/O alignment. • More robust than dynamic or differencing VHD due to the lack of block allocation tables (i.e. redirection layer). • File-based storage container has more management advantages than pass-through disk. • Expanding is available to increase the capacity of VHD. • No risk of underlying volume running out of space during VM operations	• Upfront space allocation may increase the storage cost when large of number fixed VHD are deployed. • Large fixed VHD Creation is time-consuming. • Shrinking the virtual capacity (i.e. reducing the virtual size) is not possible.

Dynamically expanding or Differencing VHD	• Good performance • Quicker to create than fixed sized VHD • Grow dynamically to save disk space and provide efficient storage usage. • Smaller VHD file size makes it more nimble in terms of transporting across the network. • Blocks of full zeros will not get allocated and thus save the space under certain circumstances. • Compact operation is available to reduce the actual physical file size.	• Interleaving of meta-data and data blocks may cause I/O alignment issues. • Write performance may suffer during VHD expanding. • Dynamically expanding and differencing VHDs cannot exceed 2040GB • May get VM paused or VHD yanked out if disk space is running out due to the dynamic growth. • Shrinking the virtual capacity is not supported. • Expanding is not available for differencing VHDs due to the inherent size limitation of parent disk. • Defrag is not recommended due to inherent re-directional layer.

Note: there are types of compaction for dynamic or differencing VHDs. One is done when file system is presented (i.e. you mount the VHD and bring the disk online and the other is not (VHD is offline). We recommend mounting a VHD first before doing compaction to improve the efficiency. This is also the default behavior when a compaction is initiated in Hyper-V Manager. However, DiskPart allows you to do an offline compaction without prompting you to mount it first. When that is the case, only data blocks full of zero will get released since no file system is involved. This may limit the effectiveness of the compact operation.Note that online compaction will fail with a file system limitation error if any volume snapshots exist.

7. Summary of supported and practical limits

Virtual Hard Disks have a lot of flexibility in formats, blocksize and usage. The following table describes some architectural limits around VHD in native use and in use by virtual machines followed by their definitions. One thing worth pointing out here is Windows Server 2008 R2 provides two different user interfaces to deal with VHDs: Hyper-V Manager and virtual disk service (VDS) based tools DiskMgmt/DiskPart. Customers may note inconsistent behavior, for example, VHD compaction as discussed previouslythe maximum size of fixed size VHDs.

Virtualization Feature	Windows Server 2008 Hyper-V(RTM)	Windows Server 2008 R2
Nested Depth (VHD in VHD)	0	2
Default VHD Block Size	512KB	2MB
Minimum VHD Size	3MB	3MB
Number of Attached VHDs	Limited by the virtual bus (64)	Limited by the number of disks Windows allows
Max Differencing Chain Length	Effective = 8 Supported = 50	Effective =1024 Supported = 1024
Max Fixed Sized VHD Size	2048GB(2040G via Hyper-V Manager)	16TB(2040G via Hyper-V Manager)
Max Dynamic VHD Size	2040GB	2040GB
Max DifferencingVHD Size	2040GB	2040GB

- Nested Depth – refers to the number of VHDs contained in VHDs that can be attached. For example nested depth 2 means a VHD is contained in a VHD that is attached.
- Default VHD Block size – This is the default size of data blocks in Dynamic and Differencing VHDs. In R2 the size was increased to improve overall throughput.
- Number of Attached VHDs – This is the number of VHDs that can be attached to the system at any one time. Keep in mind that after all drive letters are usedyou will have to use volume naming or junction points to references drives.
- MaxDifferencingVHDchain length – This defines how many virtual hard disks can be associated in a hierarchical chain of 'descendents' – starting with the original parent disk and ending with a child disk that has no children below it.

- Max fixed sized/dynamically expanding/differencingVHD Size – This is the maximum size in bytes that the format can present as a single disk.

In addition to the limits above there are some limits specific to VHD use in virtual machines that impacts overall performance. The following table lists those limits followed by their description.

Virtualization Feature	Windows Server 2008	Windows Server 2008 R2
Largest Virtual SCSI IO Size	128KB	8MBytes
Largest Virtual IDE IO Size	128KB	128KB

- Largest Virtual SCSI IO Size – This is the maximum size of and IO passed from a VM to the management partition for processing over the Virtual SCSI path. Larger I/O size may improve CPU efficiency by reducing I/O split.
- Largest Virtual IDE IO Size – This is the maximum size of IO passed from a VM to the management partition for processing over the Virtual IDE path.

If in addition to the limits above you can also see a feature comparison (including storage) between Windows Server 2008 and Windows Server 2008 R2 at:

http://blogs.msdn.com/tvoellm/archive/2009/08/05/what-s-new-in-windows-server-2008-r2-hyper-v-performance-and-scale.aspx

8. Closing

Virtual Hard Disk technology has continued to improve release-over–release, both in terms of the scenarios supported (like boot from VHD) as well as overall performance. This paper details the significant improvements in dynamically expanding and differencingVHDrandom performance, the substantiallyfaster fixed sized VHD creation speed, and almost zero percent drop in throughput to large differencing VHD chains due to better caching.

When choosing the right VHD for your environment you should consider both the access performance and storage needs. With the improvements demonstrated in Windows Server 2008 R2 the choice has less to do with the access speed and more to do with the amount of memory used due to advanced caching.

9. Acknowledgements

We want to thank the following people and the entire VHD team for the contribution and hard work that were incorporated into the VHD performanceimprovement in Windows Server 2008 R2:

Dustin Green, Todd Harris, Mike Ebersol, Karthik Thirumalai, John Howard, Chris Eck, Taylor Brown

10. References

- Loopback mounting of VHDs - http://blogs.msdn.com/virtual_pc_guy/archive/2008/02/01/mounting-a-virtual-hard-disk-with-hyper-v.aspx
- Inbox APIs (virtdisk.dll) - http://msdn.microsoft.com/en-us/magazine/dd569754.aspx
- VHD Specification- http://technet.microsoft.com/en-us/virtualserver/bb676673.aspx
- VHD FAQ - http://technet.microsoft.com/en-us/library/dd440865(WS.10).aspx
- What's new in Hyper-V in Windows Server 2008 R2 - http://www.microsoft.com/windowsserver2008/en/us/hyperv-r2.aspx