Performance Characterization of VMFS and RDM Using a SAN

ESX Server 3.5

VMware® ESX Server offers two choices for managing disk access in a virtual machine—VMware Virtual Machine File System (VMFS) and raw device mapping (RDM). It is very important to understand the I/O characteristics of these disk access management systems in order to choose the right access type for a particular application. Choosing the right disk access management method can be a key factor in achieving high system performance for enterprise-class applications.

This paper is a follow-on to a previous performance study that compares the performance of VMFS and RDM in ESX Server 3.0.1 ("Performance Characteristics of VMFS and RDM: VMWare ESX Server 3.0.1";see "Resources" on page 11 for a link). The experiments described in this paper compare the performance of VMFS and RDM in VMware ESX Server 3.5. The goal is to provide data on performance and system resource utilization at various load levels for different types of workloads. This information offers you an idea of relative throughput, I/O rate, and CPU cost for each of the options so you can select the appropriate disk access method for your application.

A direct comparison of the results in this paper to those reported in the previous paper would be inaccurate. The test setup we used to conduct tests for this paper is different from the one used for the tests described in the previous paper with ESX Server 3.0.1. Previously, we created the test disks on local 10K rpm SAS disks. In this paper we used Fibre Channel disks in an EMC CLARiiON CX3-40 to create the test disk. Because of the different protocols used to access the disks, the I/O path in the ESX Server software stack changes significantly and thus the I/O latency experienced by the workloads in `iometer` also change.

This study covers the following topics:

Technology Overview

VMFS is a special high-performance file system offered by VMware to store ESX Server virtual machines. Available as part of ESX Server, it is a clustered file system that allows concurrent access by multiple hosts to files on a shared VMFS volume. VMFS offers high I/O capabilities for virtual machines. It is optimized for storing and accessing large files such as virtual disks and the memory images of suspended virtual machines.

RDM is a mapping file in a VMFS volume that acts as a proxy for a raw physical device. The RDM file contains metadata used to manage and redirect disk accesses to the physical device. This technique provides advantages of direct access to physical device in addition to some of the advantages of a virtual disk on VMFS storage. In brief, it offers VMFS manageability with the raw device access required by certain applications.

You can configure RDM in two ways:

- Virtual compatibility mode—This mode fully virtualizes the mapped device, which appears to the guest operating system as a virtual disk file on a VMFS volume. Virtual mode provides such benefits of VMFS as advanced file locking for data protection and use of snapshots.
- Physical compatibility mode—This mode provides access to most hardware characteristics of the mapped device. VMkernel passes all SCSI commands to the device, with one exception, thereby exposing all the physical characteristics of the underlying hardware.

Both VMFS and RDM provide such clustered file system features as file locking, permissions, persistent naming, and VMotion capabilities. VMFS is the preferred option for most enterprise applications, including databases, ERP, CRM, VMware Consolidated Backup, Web servers, and file servers. Although VMFS is recommended for most virtual disk storage, raw disk access is needed in a few cases. RDM is recommended for those cases. Some of the common uses of RDM are in cluster data and quorum disks for configurations using clustering between virtual machines or between physical and virtual machines or for running SAN snapshot or other layered applications in a virtual machine.

For more information on VMFS and RDM, see the Server Configuration Guide mentioned in "Resources" on page 11.

Executive Summary

The main conclusions that can be drawn from the tests described in this study are:

- For random reads and writes, VMFS and RDM yield a similar number of I/O operations per second.
- For sequential reads and writes, performance of VMFS is very close to that of RDM (except on sequential reads with an I/O block size of 4K). Both RDM and VMFS yield a very high throughput in excess of 300 megabytes per second depending on the I/O block size
- For random reads and writes, VMFS requires 5 percent more CPU cycles per I/O operation compared to RDM.
- For sequential reads and writes, VMFS requires about 8 percent more CPU cycles per I/O operation compared to RDM.

Test Environment

The tests described in this study characterize the performance of VMFS and RDM in ESX Server 3.5. We ran the tests with a uniprocessor virtual machine using Windows Server 2003 Enterprise Edition with SP2 as the guest operating system. The virtual machine ran on an ESX Server system installed on a local SCSI disk. We attached two disks to the virtual machine—one virtual disk for the operating system and a separate test disk, which was the target for the I/O operations. We generated I/O load using Iometer, a very popular tool for evaluating I/O performance (see "Resources" on page 11 for a link to more information). See "Configuration" on page 10 for a detailed list of the hardware and software configuration we used for the tests.

For all workloads except 4K sequential read, we used the default cache page size (8K) in the storage. However, for 4K sequential read workloads, the default cache page setting resulted in lower performance for both VMFS and RDM. EMC recommends setting the cache page size in storage to application block size (4K in this case)

for a stable workload with a constant I/O block size. Hence, for 4K sequential read workloads, we set the cache page size to 4K. See Figure 12 and Figure 13 for a performance comparison of 4K sequential read I/O operations with 4K and 8K cache page settings.

For more information on cache page settings, see the white paper "EMC CLARiiON Best Practices for Fibre Channel Storage" white paper (see "Resources" on page 11 for a link).

Disk Layout

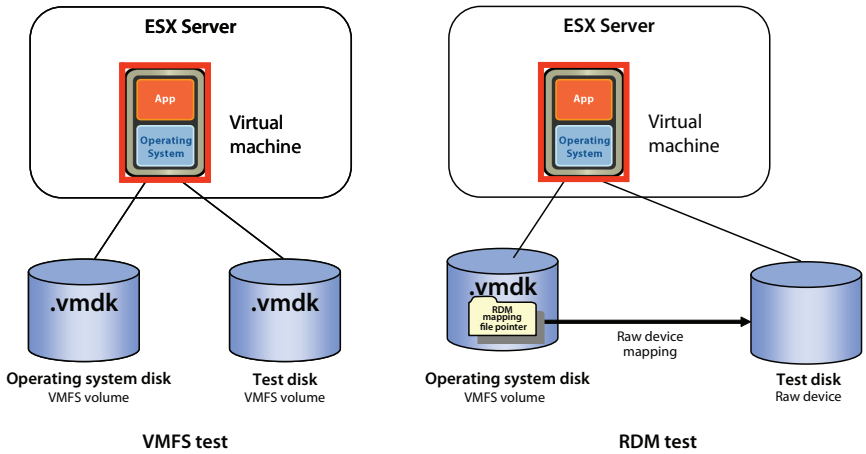
In this study, disk layout refers to the configuration, location, and type of disk used for tests. We used a single server attached to a Fibre Channel array through a host bus adapter for the tests.

We created a logical drive on a single physical disk. We installed ESX Server on this drive. On the same drive, we also created a VMFS partition and used it to store the virtual disk on which we installed the Windows Server 2003 guest operating system.

The test disk was located on a metaLUN on the CLARiiON CX3-40, which was created as follows: We created two RAID 0 groups on the CLARiiON CX3-40, one containing 15 Fibre Channel disks and the other containing 10 Fiber Channel disks. We configured a 10GB LUN on each RAID group. We then created a 20GB metaLUN using the two 10GB LUNs in a striped configuration (see "Resources" on page 11 for a link to the *EMC Navisphere Manager Administrator's Guide* to get more information). We used a metaLUN for the test disk because it allowed us to use more than 16 spindles to stripe a LUN in a RAID 0 configuration. We used this test disk only for the I/O stress test. We created a virtual disk on the test disk and attached that virtual disk to the Windows virtual machine. To the guest operating system, the virtual disk appears as a physical drive.

Figure 1 shows the disk configuration used in our tests. In the VMFS tests, we implemented the virtual disk (seen as a physical disk by the guest operating system) as a `.vmdk` file stored on a VMFS partition created on the test disk. In the RDM tests, we created an RDM file on the VMFS volume (the volume that held the virtual machine configuration files and the virtual disk where we installed the guest operating system) and mapped the RDM file to the test disk. We configured the test virtual disk so it was connected to an LSI SCSI host bus adapter.

Figure 1. Disk layout for VMFS and RDM tests



From the perspective of the guest operating system, the test disks were raw disks with no partition or file system (such as NTFS) created on them. Iometer can read and write raw, unformatted disks directly. We used this capability so we could compare the performance of the underlying storage implementation without involving any operating system file system.

Software Configuration

We configured the guest operating system to use the LSI Logic SCSI driver. On VMFS volumes, we created virtual disks with the `thick` option. This option provides the best-performing disk allocation scheme for a virtual disk. All the space allocated during disk creation is available for the guest operating system

immediately after the creation. Any old data that might be present on the allocated space is not zeroed out during virtual machine write operations.

Unless stated otherwise, we left all ESX Server and guest operating system parameters at their default settings. In each test case, we zeroed the virtual disks before starting the experiment using the command-line program `vmkfstools` (with the `–w` option)

NOTE ESX Server 3.5 offers four options for creating virtual disks—`zeroedthick`, `eagerzeroedthick`, `thick`, and `thin`. When a virtual disk is created using the VI Client, the `zeroedthick` option is used by default. Virtual disks with `eagerzeroedthick`, `thick`, or `thin` formats can be created only with `vmkfstools`, a command-line program. The `zeroedthick` and `thin` formats have characteristics similar to the `thick` format after the initial write operation to the disk. In our tests, we used the `thick` option to prevent the "warm-up" anomalies. For details on the supported virtual disk formats refer to Chapter 5 of the VMware Infrastructure 3 *Server Configuration Guide*. For details on using `vmkfstools`, see appendixes of the VMware Infrastructure 3 *Server Configuration Guide*.

I/O Workload Characteristics

Enterprise applications typically generate I/O with mixed access patterns. The size of data blocks transferred between the server hosting the application and the storage also changes. Designing an appropriate disk and file system layout is very important to achieve optimum performance for a given workload.

A few applications have a single access pattern. One example is backup and its pattern of sequential reads. Online transaction processing (OLTP) database access, on the other hand, is highly random. The nature of the application also affects the size of data blocks transferred. Often, the data block size is not a single value but a range.

The I/O characteristics of a workload can be defined in terms of the ratio of read to write operations, the ratio of sequential to random I/O access, and the data transfer size.

Test Cases

In this study, we characterize the performance of VMFS and RDM for a range of data transfer sizes across various access patterns. The data transfer sizes we selected were 4KB, 8KB, 16KB, 32KB, and 64KB. The access patterns we chose were random reads, random writes, sequential reads, sequential writes, or a mix of random reads and writes. The test cases are summarized in Table 1.

Table 1. Test cases

	100% Sequential	100% Random
100% Read	4KB, 8KB, 16KB, 32KB, 64KB	4KB, 8KB, 16KB, 32KB, 64KB
100% Write	4KB, 8KB, 16KB, 32KB, 64KB	4KB, 8KB, 16KB, 32KB, 64KB
50% Read + 50% Write		4KB, 8KB, 16KB, 32KB, 64KB

Load Generation

We used the Iometer benchmarking tool, originally developed at Intel and widely used in I/O subsystem performance testing, to generate I/O load and measure the I/O performance. For a link to more information, see "Resources" on page 11. Iometer provides options to create and execute a well-designed set of I/O workloads. Because we designed our tests to characterize the relative performance of virtual disks on raw devices and VMFS, we used only basic load emulation features in the tests.

Iometer configuration options used as variables in the tests:

- Transfer request sizes: 4KB, 8KB, 16KB, 32KB, and 64KB.
- Percent random or sequential distribution: for each transfer request size, we selected 0 percent random access (equivalent to 100 percent sequential access) and 100 percent random accesses.
- Percent read or write distribution: for each transfer request size, we selected 0 percent read access (equivalent to 100 percent write access), 100 percent read accesses, and 50 percent read access/50 percent

write access (only for random access, which is referred to as "random mixed" workload in the rest of the paper).

Iometer parameters constant for all test cases:

- Number of outstanding I/O operations: 64
- Runtime: 5 minutes
- Ramp-up time: 60 seconds
- Number of workers to spawn automatically: 1

Performance Results

This section presents data and analysis of storage subsystem performance in a uniprocessor virtual machine.

Metrics

The metrics we used to compare the performance of VMFS and RDM are I/O rate (measured as number of I/O operations per second), throughput rate (measured as MB per second), and CPU cost measured in terms of MHz per I/Ops.

In this study, we report the I/O rate and throughput rate as measured by Iometer. We use a cost metric measured in terms of MHz per I/Ops to compare the efficiencies of VMFS and RDM. This metric is defined as the CPU cost (in processor cycles) per unit I/O and is calculated as follows:

$$\text{MHz per I/Ops} = \frac{\text{Average CPU utilization} \times \text{CPU rating in MHz} \times \text{Number of cores}}{\text{Number of I/O operations per second}}$$

We collected I/O and CPU utilization statistics from Iometer and `esxtop` as follows:

- Iometer—collected I/O operations per second and throughput in MBps
- `esxtop`—collected the average CPU utilization of physical CPUs

For links to additional information on how to collect I/O statistics using Iometer and how to collect CPU statistics using `esxtop`, see "Resources" on page 11.

Performance

This section compares the performance characteristics of each type of disk access management. The metrics used are I/O rate, throughput, and CPU cost.

Random Workload

In our tests for random workloads, VMFS and RDM produced similar I/O performance as evident from Figure 2, Figure 3, and Figure 4.

Figure 2. Random mixed (50 percent read access/50 percent write access) I/O operations per second (higher is better)

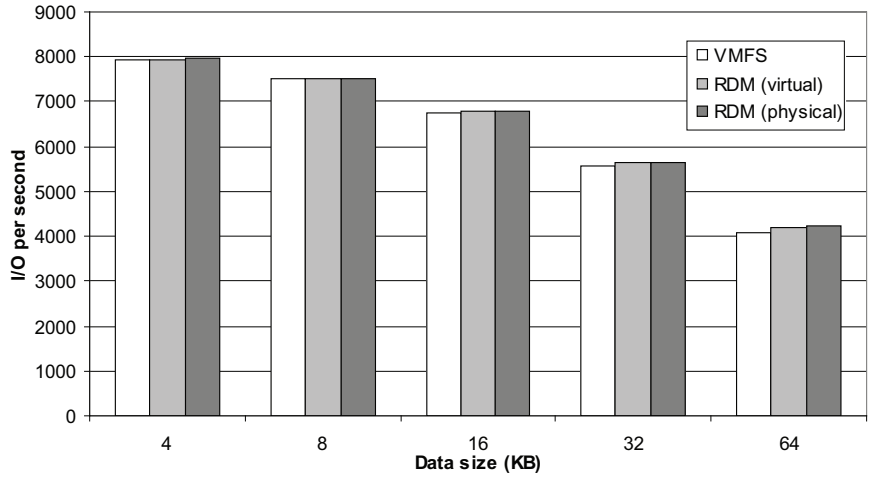


Figure 3. Random read I/O operations per second (higher is better)

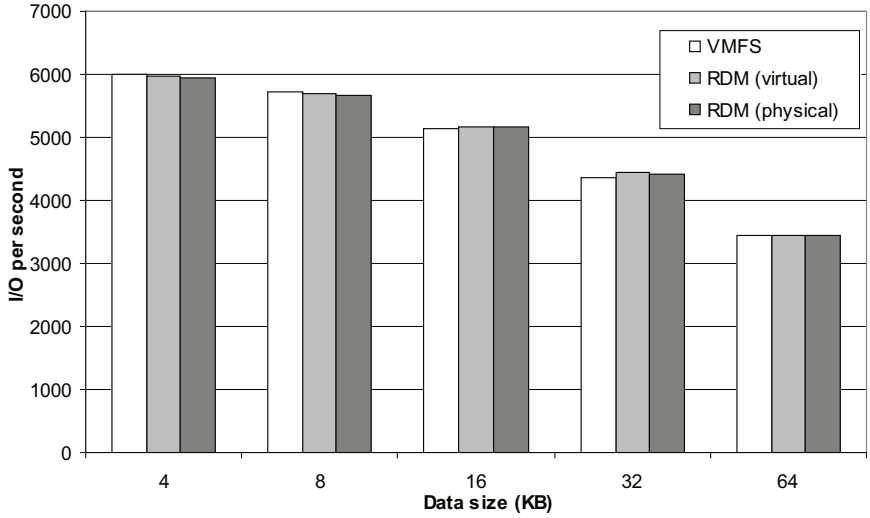
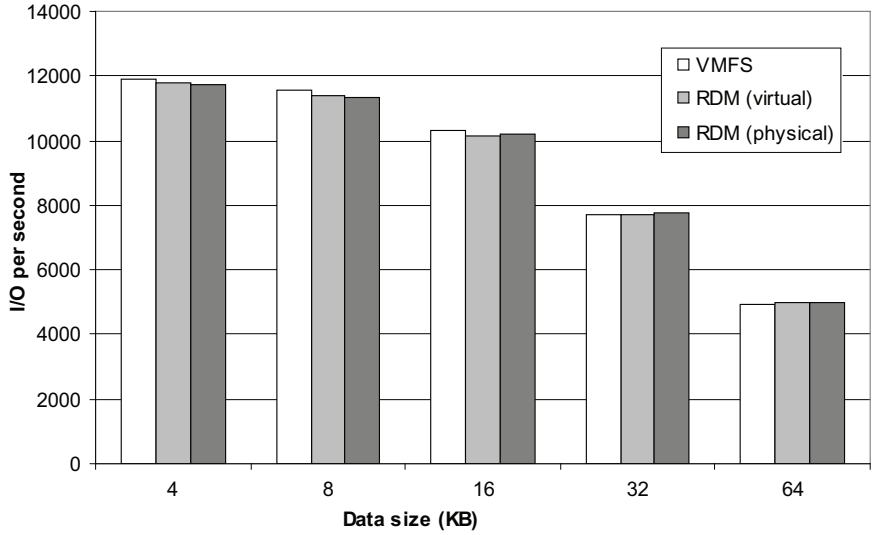


Figure 4. Random write I/O operations per second (higher is better)



Sequential Workload

For 4K sequential read, we changed the cache page size on the CLARiiON CX3-40 to 4K. We used the default size of 8K for all other workloads.

In ESX Server 3.5, for sequential workloads, performance of VMFS is very close to that of RDM for all I/O block sizes except 4K sequential read. Most applications with a sequential read I/O pattern use a block size greater than 4K. Both VMFS and RDM provide similar performance in those cases, as shown in Figure 5 and Figure 6.

Both VMFS and RDM deliver very high throughput (in excess of 300 megabytes per second, depending on the I/O block size).

Figure 5. Sequential read I/O operations per second (higher is better)

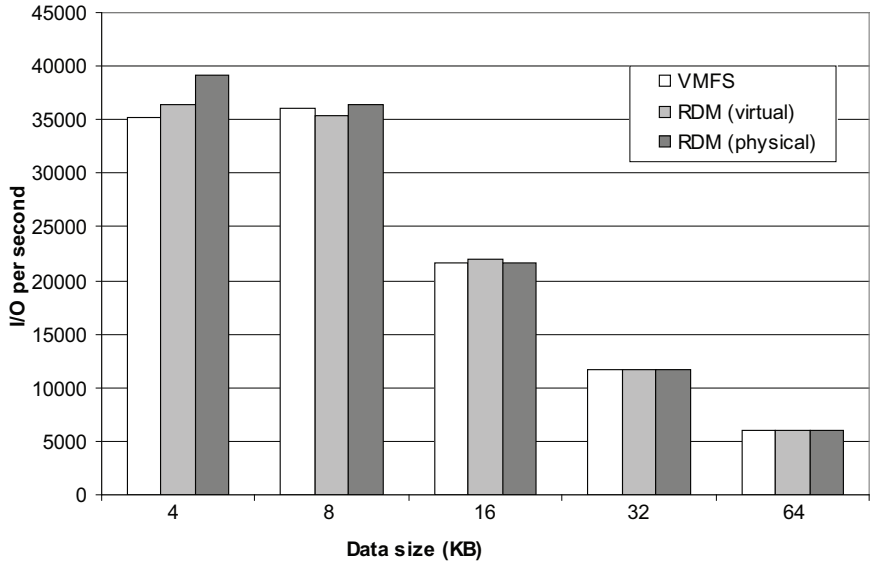


Figure 6. Sequential write I/O operations per second (higher is better)

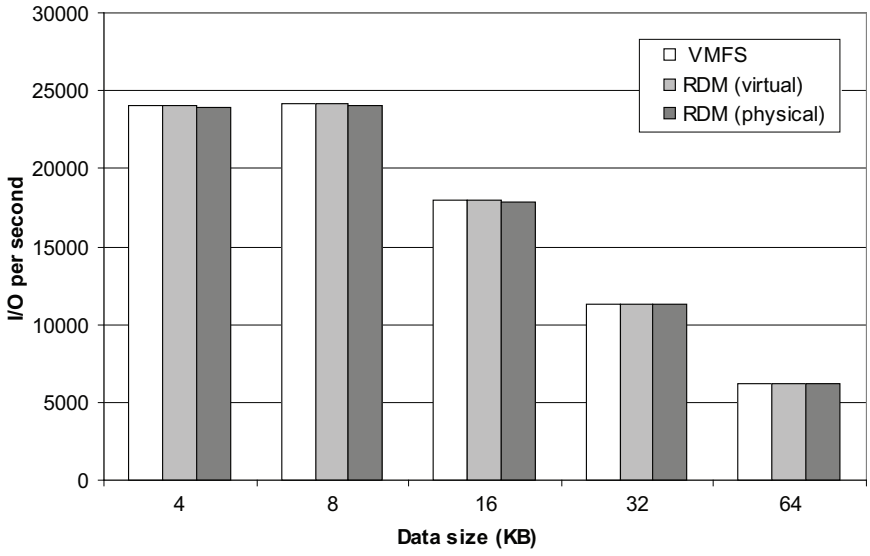


Table 2 and Table 3 show the throughput rates in megabytes per second corresponding to the above I/O operations per second for VMFS and RDM. The throughput rates (I/O operations per second * data size) are consistent with the I/O rates shown above and display behavior similar to that explained for the I/O rates.

Table 2. Throughput rates for random workloads in megabytes per second

	Random Mix			Random Read			Random Write		
Data Size (KB)	VMFS	RDM (V)	RDM (P)	VMFS	RDM (V)	RDM (P)	VMFS	RDM (V)	RDM (P)
4	30.96	30.98	31.15	23.41	23.27	23.27	46.38	46.1	45.87
8	58.8	58.68	58.68	44.67	44.43	44.35	90.24	89.22	88.74
16	105.22	106.03	105.09	80.14	80.72	80.58	161.2	158.25	159.42
32	173.72	176.1	176.69	135.91	138.53	138.11	241.4	241.12	242.05
64	256.14	262.92	263.98	215.1	215.49	214.59	309.6	311.13	310.15

Table 3. Throughput rates for sequential workloads

	Sequential Read			Sequential Write		
Data Size (KB)	VMFS	RDM (V)	RDM (P)	VMFS	RDM (V)	RDM (P)
4	137.21	142.13	153	93.76	93.8	93.36
8	272.61	276.35	284.41	188.56	188.45	187.84
16	341.08	342.41	338.75	280.95	281.17	278.91
32	363.86	365.26	364.23	352.17	354.86	352.89
64	377.35	377.6	377.09	384.02	385.36	386.81

CPU Cost

CPU cost can be computed in terms of CPU cycles required per unit of I/O or unit of throughput (byte). We obtained a figure for CPU cycles used by the virtual machine for managing the workload, including the virtualization overhead, by multiplying the average CPU utilization of all the processors seen by ESX Server, the CPU rating in MHz, and the total number of cores in the system (four in our test server). In this study, we measured CPU cost as CPU cycles per unit of I/O operations per second.

Normalized CPU cost for various workloads is shown in figures 7 through 11. We used CPU cost for RDM (physical mapping) as the baseline for each workload and plotted the CPU costs of VMFS and RDM (virtual mapping) as a fraction of the baseline value. For random workloads the CPU cost of VMFS is on average 5 percent more than the CPU cost of RDM. For sequential workloads, the CPU cost of VMFS is 8 percent more than the CPU cost of RDM.

As with any file system, VMFS maintains data structures that map filenames to physical blocks on the disk. Each file I/O requires accessing the metadata to resolve filenames to actual data blocks before reading data from or writing data to a file. The address resolution requires a few extra CPU cycles every time there is an I/O access. In addition, maintaining the metadata also requires additional CPU cycles. RDM does not require any underlying file system to manage its data. Data is accessed directly from the disk, without any file system overhead, resulting in a lower CPU cycle consumption and better CPU cost for RDM access.

Figure 7. Normalized CPU cost for random mix (lower is better)

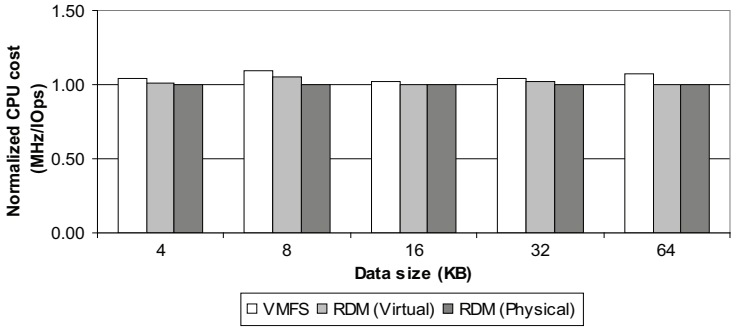


Figure 8. Normalized CPU cost for random read (lower is better)

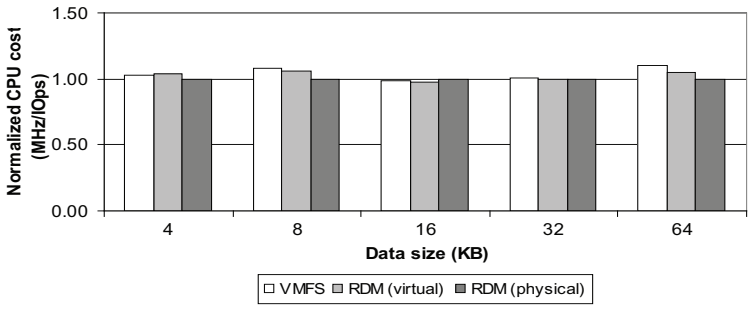


Figure 9. Normalized CPU cost for random write (lower is better)

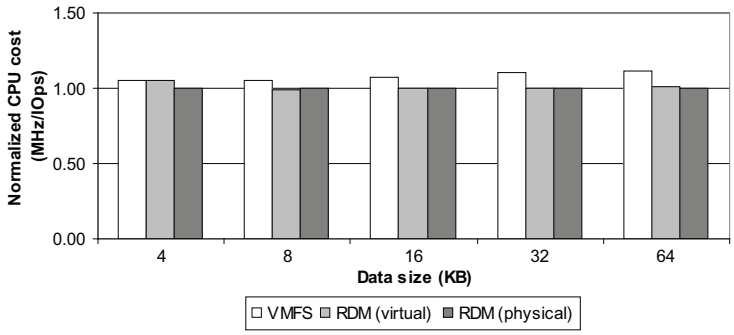


Figure 10. Normalized CPU cost for sequential read (lower is better)

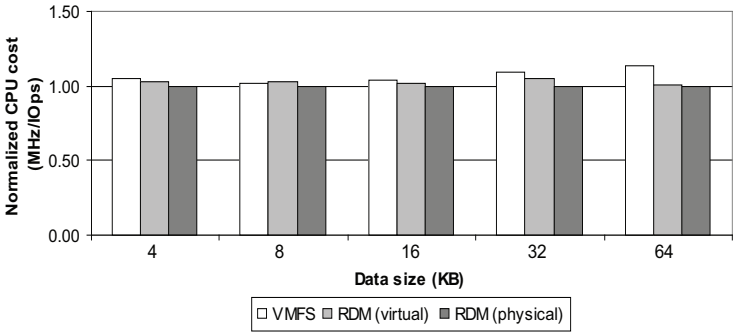
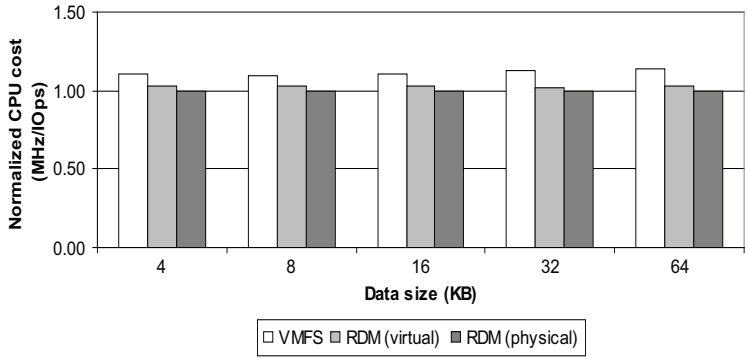


Figure 11. Normalized CPU cost for sequential write (lower is better)



Conclusion

VMware ESX Server offers two options for disk access management—VMFS and RDM. Both options provide clustered file system features such as user-friendly persistent names, distributed file locking, and file permissions. Both VMFS and RDM allow you to migrate a virtual machine using VMotion. This study compares the performance characteristics of both options and finds only minor differences in performance.

For random workloads, VMFS and RDM produce similar I/O throughput. For sequential workloads with small I/O block sizes, RDM provides a small increase in throughput compared to VMFS. However, the performance gap decreases as the I/O block size increases. For all workloads, RDM has slightly better CPU cost.

The test results described in this study show that VMFS and RDM provide similar I/O throughput for most of the workloads we tested. The small differences in I/O performance we observed were with the virtual machine running CPU-saturated. The differences seen in these studies would therefore be minimized in real life workloads because most applications do not usually drive virtual machines to their full capacity. Most enterprise applications can, therefore, use either VMFS or RDM for configuring virtual disks when run in a virtual machine.

However, there are a few cases that require use of raw disks. Backup applications that use such inherent SAN features as snapshots or clustering applications (for both data and quorum disks) require raw disks. RDM is recommended for these cases. We recommend use of RDM for these cases not for performance reasons but because these applications require lower-level disk control.

Configuration

This section describes the hardware and software configurations we used in the tests described in this study.

Server Hardware

- Server: Dell PowerEdge 2950
- Processors: 2 dual-core Intel Xeon 5160 processors, 3.00GHz, 4MB L2 cache (4 cores total)
- Memory: 8GB
- Local disks: 1 Seagate 146GB 10K RPM SAS (for ESX Server and the guest operating system)

Storage Hardware

- Storage: CLARiiON CX3-40 (4Gbps)
- Memory: 4GB per storage processor
- Fiber channel disks: 25 Seagate 146GB 15K RPM in RAID 0 configuration (first five disks containing Flare OS were not used)
- HBA: QLA 2460 (4Gbps)

Software

- Virtualization software: ESX Server 3.5 (build 64607)

Guest Operating System Configuration

- Operating system: Windows Server 2003 R2 Enterprise Edition 32-bit, Service Pack 2, 512MB of RAM, 1 CPU
- Test disk: 20GB unformatted disk

Storage Configuration

- Read cache: 1GB per storage processor
- Write cache: 1.9GB
- RAID-0 group 1: 10 disks (10GB LUN)
- RAID-0 group 2: 15 disks (10GB LUN)
- MetaLUN: 20GB

Iometer Configuration

- Number of outstanding I/Os: 64
- Ramp-up time: 60 seconds
- Run time: 5 minutes
- Number of workers (threads): 1
- Access patterns: random/mix, random/read, random/write, sequential/read, sequential/write
- Transfer request sizes: 4KB, 8KB, 16KB, 32KB, 64KB

Resources

- "Performance Characteristics of VMFS and RDM: VMWare ESX Server 3.0.1"
  http://www.vmware.com/resources/techresources/1019
- To obtain Iometer, go to
  http://www.iometer.org/
- For more information on how to gather I/O statistics using Iometer, see the Iometer user's guide at
  http://www.iometer.org/doc/documents.html
- To learn more about how to collect CPU statistics using `esxtop`, see "Using the esxtop Utility" in the VMware Infrastructure 3 Resource Management Guide at
  http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_resource_mgmt.pdf
- For a detailed description of VMFS and RDM and how to configure them, see chapters 5 and 8 of the VMware Infrastructure 3 *Server Configuration Guide* at
  http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_3_server_config.pdf
- VMware Infrastructure 3 *Server Configuration Guide*
  http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_3_server_config.pdf
- "EMC CLARiiON Best Practices for Fibre Channel Storage"
  http://powerlink.emc.com
- EMC *Navisphere Manager Administrator's Guide*
  http://powerlink.emc.com

Appendix: Effect of Cache Page Size on Sequential Read I/O Patterns with I/O Block Size Less than Cache Page Size

The default cache page setting on a CLARiiON CX3-40 is 8K. This setting affects sequential reads with a block size smaller than the cache page size (refer to "EMC CLARiiON Best Practices for Fibre Channel Storage" mentioned in "Resources" on page 11). Following the EMC recommendation, we changed the cache page setting to 4K, which was same as the I/O block size generated by Iometer. This resulted in a 226 percent increase in the number of I/O operations per second as shown in Figure 12.

Figure 12. Sequential read I/O operations per second for 4k sequential read with different cache page size (higher is better)

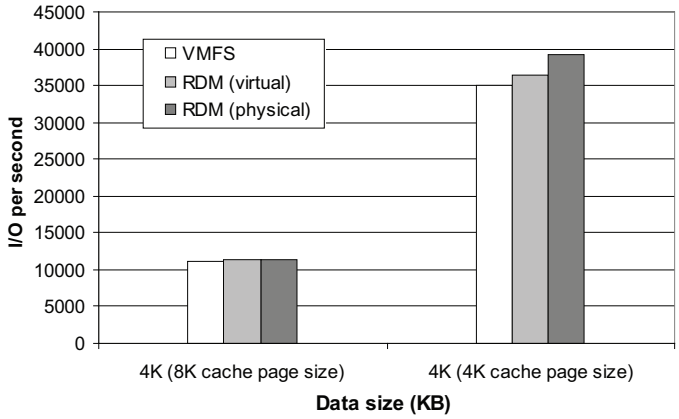
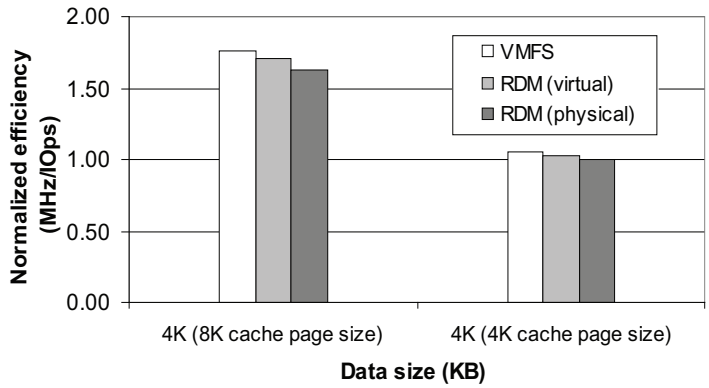


Figure 13 shows the normalized CPU costs per I/O operation for VMFS and RDM with 4K and 8K cache page size. We used the CPU cost of RDM (physical mapping) with 4K cache page size as the baseline value and plotted the CPU costs of VMFS and RDM (both physical and virtual mapping) with both 4K and 8K cache page size as a fraction of the baseline value. The CPU cost improved by an average value of 70 percent with 4K cache page size for both VMFS and RDM

Figure 13. Normalized CPU cost for 4K sequential read with different cache page size (lower is better)



VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Revision 20080131 Item: PS-051-PRD-01-01