Performance Characteristics of VMFS and RDM

VMware ESX Server 3.0.1

VMware ESX Server offers three choices for managing disk access in a virtual machine—VMware Virtual Machine File System (VMFS), virtual raw device mapping (RDM), and physical raw device mapping. 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 study provides performance characterization of the various disk access management methods supported by VMware ESX Server. The goal is to provide data on performance and system resource utilization at various load levels for different types of work loads. This information offers you an idea of relative throughput, I/O rate, and CPU efficiency for each of the options so you can select the appropriate disk access method for your application.

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 distributed 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 minimal SCSI virtualization 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 distributed 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, refer to the *Server Configuration Guide*.

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 similar I/O performance.
- For sequential reads and writes, RDM provides slightly better I/O performance at smaller I/O block sizes, but VMFS and RDM provide similar I/O performance for larger I/O block sizes (greater than 32KB).
- CPU cycles required per I/O operation for both VMFS and RDM are similar for random reads and writes at smaller I/O block sizes. As the I/O block size increases, CPU cycles per I/O operation increases for VMFS compared to RDM.
- For sequential reads and writes, RDM requires relatively fewer CPU cycles per I/O operation.

Test Environment

The tests described in this study characterize the performance of various options for disk access management VMware offers with ESX Server. 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 used a logical volume created on five local SCSI disks configured as RAID 0 as the test disk. 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.

Disk Layout

In this study, disk layout refers to the configuration, location, and type of disk used for tests. We used a single server with six physical SAS disks and a RAID controller for the tests.

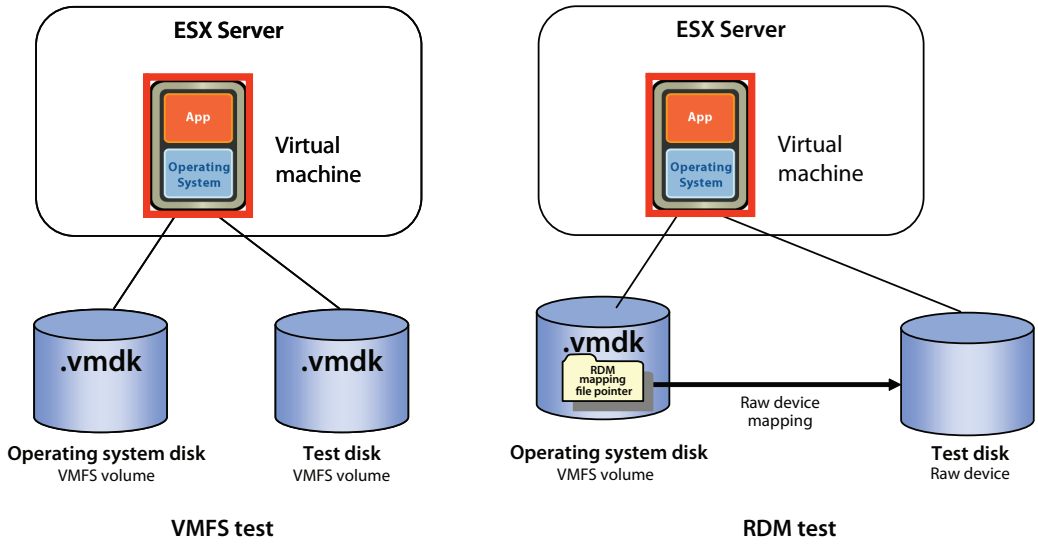
We created a logical drive on a single physical disk. We installed ESX Server on this drive. On the same drive, we also created a VMFS partition and used it to store the virtual disk on which we installed the Windows Server 2003 guest operating system.

We configured the remaining five physical disks as RAID 0 and created a 10GB logical volume, referred to here as the test disk. 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.

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.

NOTE The `thick` option is not the default when you create a virtual disk using the VI Client. To create virtual disks with the `thick` option, we used a command-line program (`vmkfstools`). For details on using vmkfstools, see 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. For Microsoft Exchange, the I/O operations are generally small—from 4KB to 16KB. OLTP database accesses using Microsoft SQL Server and Oracle databases use a mix of random 8KB read and write operations.

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. A range of data transfer sizes can also be mentioned instead of a single value.

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), 50 percent read access (only for random access) and 100 percent read accesses.

Iometer parameters constant for all test cases:

- Number of outstanding I/O operations: 8
- 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 efficiency.

In this study, we report the I/O rate and throughput rate as measured by Iometer. We use an efficiency metric called 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 efficiency.

Random Workload

Most of the enterprise applications that generate random workloads typically transfer small amounts of data in each I/O operation. For a small I/O data transfer, the time it takes for the disk head to physically locate the data on the disk and move towards it is much greater than the time to actually read or write the data. Any delay caused by the overhead in the application and file system layer becomes negligible compared to the seek delay. 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 /IO per second (higher is better)

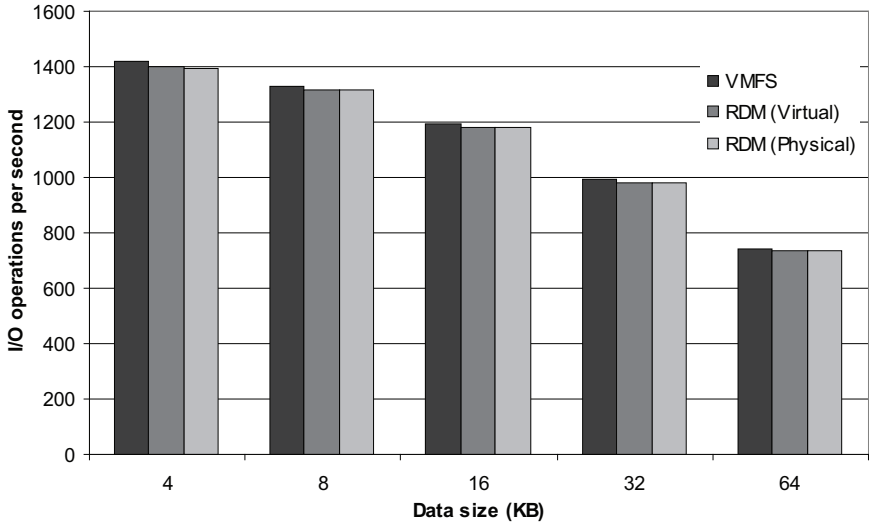


Figure 3. Random read I/O operations per second (higher is better)

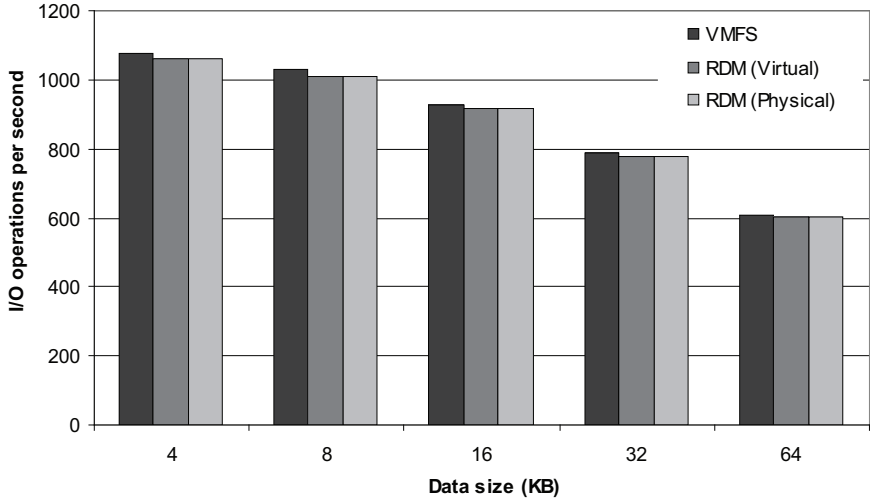
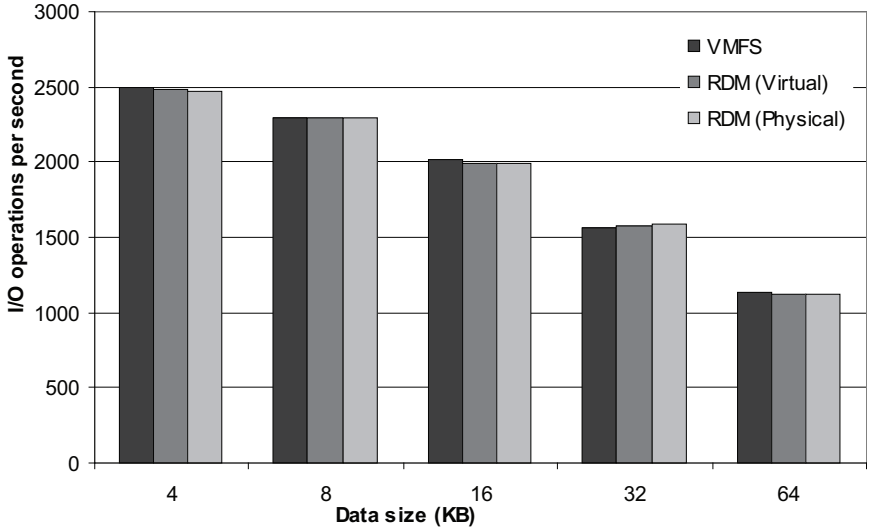


Figure 4. Random write I/O operations per second (higher is better)



Sequential Workload

For sequential workloads, applications typically access consecutive data blocks. The disk seek time is reduced because the disk head has to move very little to access the next block. Any I/O queuing delay that occurs in the application and file system layer can affect the overall throughput. Accessing file metadata and data structures to resolve file names to physical data blocks on disks adds a slight delay to the overall time required to complete the I/O operation. In order to saturate the I/O bandwidth, which typically is a few hundred megabytes per second, applications have to issue more I/O requests if the I/O block size is small. As the number of I/O requests increases, the delay caused by file system overhead increases. Thus raw disks yield slightly higher throughput compared to file systems at smaller data transfer sizes. In the sequential workload tests with a small I/O block size, RDM provided 3 to 10 percent higher I/O throughput, as shown in Figure 5 and Figure 6.

However, as I/O block size increases, the number of I/O requests decreases because the I/O bandwidth is saturated, and the delay caused by file system overhead becomes less significant. In this case, both RDM and VMFS yield similar performance.

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

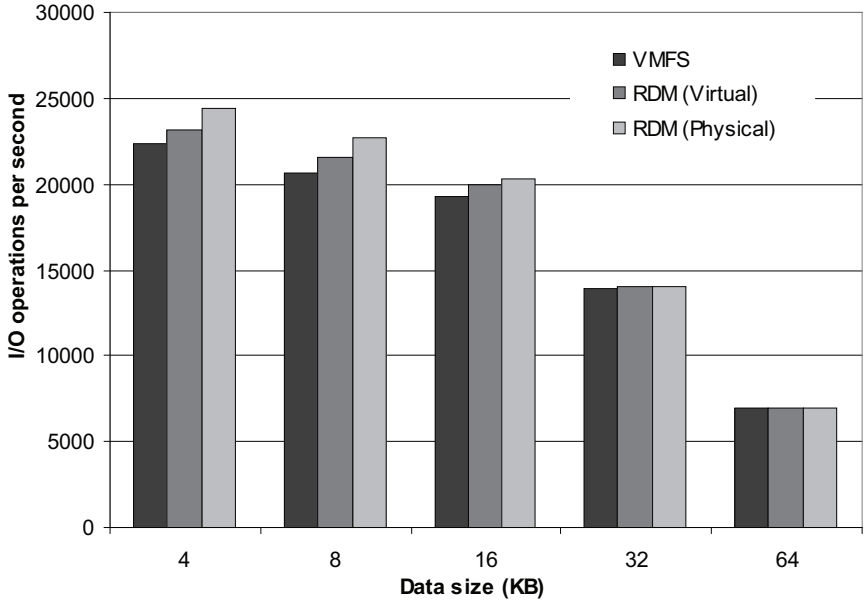


Figure 6. Sequential write I/O operations per second (higher is better)

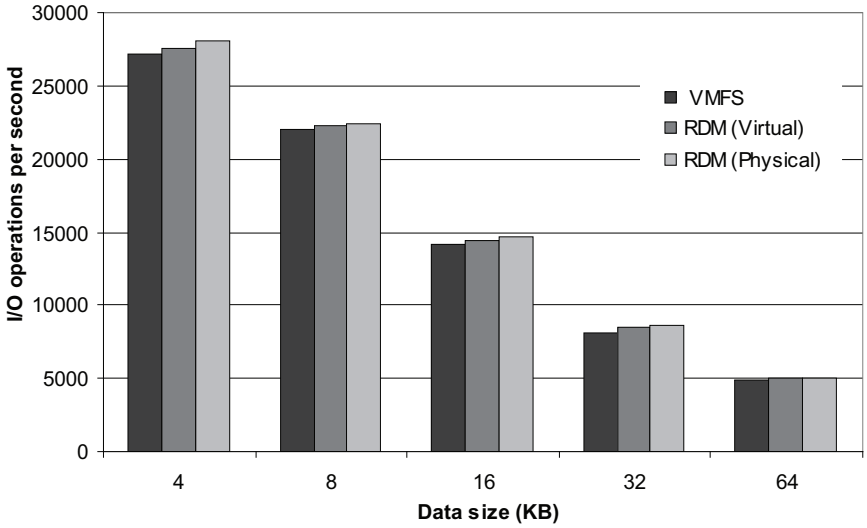


Table 2 and Table 3 show the throughput rates in megabytes per second corresponding to the above I/O operations per second for VMFS and RDM. The throughput rates (I/O operations per second * data size) are consistent with the I/O rates shown above and display behavior similar to that explained for the 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	5.54	5.46	5.45	4.2	4.14	4.15	9.78	9.7	9.64
8	10.39	10.28	10.27	8.03	7.9	7.9	17.95	17.95	17.92
16	18.63	18.43	18.43	14.49	14.3	14.34	31.58	31.2	31.06
32	31.04	30.65	30.65	24.7	24.39	24.38	50	49.24	49.45
64	46.6	45.8	45.78	38.11	37.69	37.66	71	70.15	70.01

Table 3. Throughput rates for sequential workloads in megabytes per second

	Sequential Read			Sequential Write		
Data Size (KB)	**VMFS**	**RDM (V)**	**RDM (P)**	**VMFS**	**RDM (V)**	**RDM (P)**
4	87.53	90.66	95.17	106.14	107.47	109.56
8	161.04	168.47	177.44	172.38	173.81	174.86
16	300.44	311.53	316.92	221.8	226.21	229.05
32	435.92	437.94	437.47	252.74	265.89	271.2
64	436.88	438.04	438.06	308.3	315.63	317

CPU Efficiency

CPU efficiency 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 this case). In this study, we measured CPU efficiency as CPU cycles per unit of I/O operations per second.

CPU efficiency for various workloads is shown in figures 7 through 11. For random workloads with smaller I/O block sizes (4KB and 8KB) CPU efficiency observed with both VMFS and RDM is very similar. For I/O block sizes greater than 8KB, RDM offers some improvement (7 to 12 percent). For sequential workloads, RDM offers an 8 to 12 percent improvement. As with any file system, VMFS maintains data structures that map file names

to physical blocks on the disk. Each file I/O requires accessing the metadata to resolve file names to actual data blocks before reading data from or writing data to a file. The name 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 higher CPU efficiency.

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

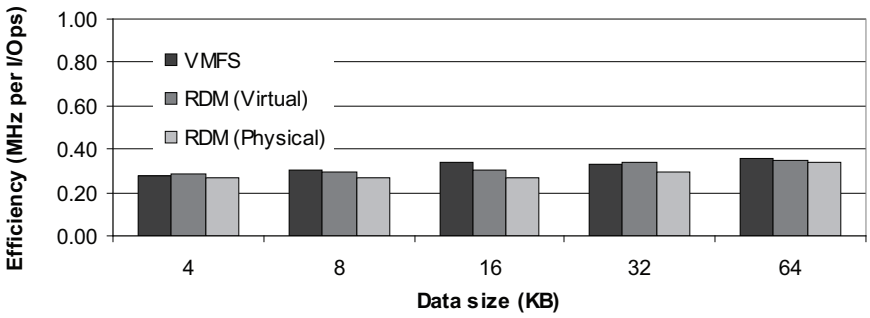


Figure 8. CPU efficiency for random read (lower is better)

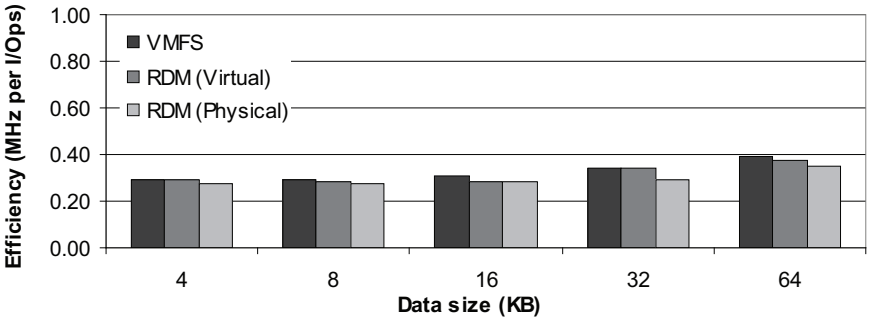


Figure 9. CPU efficiency for random write (lower is better)

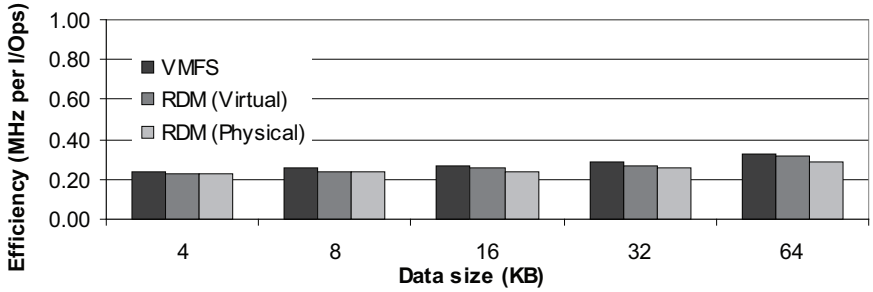


Figure 10. CPU efficiency for sequential read (lower is better)

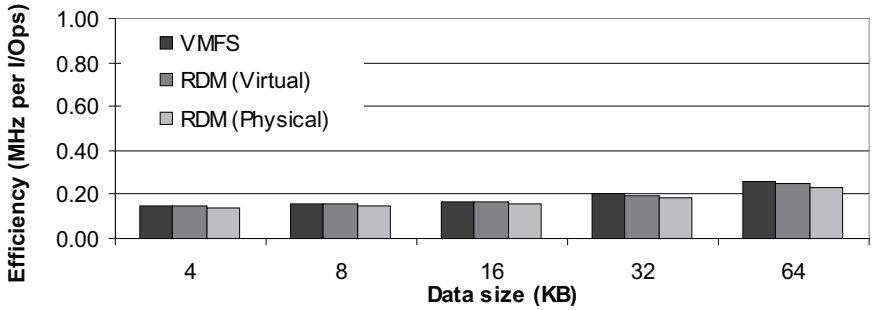
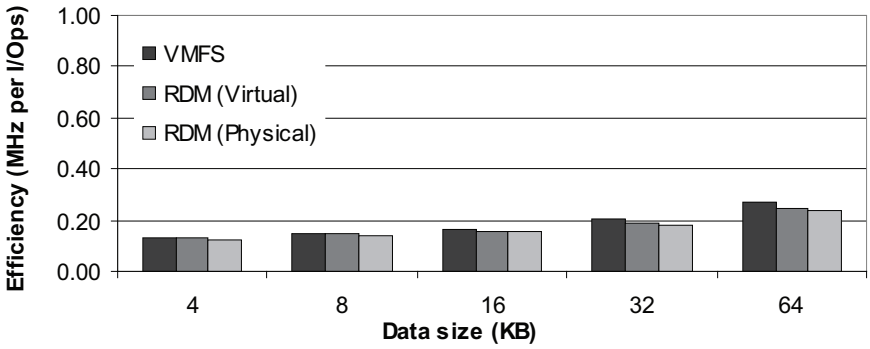


Figure 11. CPU efficiency for sequential write (lower is better)



Conclusion

VMware ESX Server offers three options for disk access management—VMFS, RDM (Virtual), and RDM (Physical). All the options provide distributed file system features like 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 three options and finds similar performance for all three.

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. VMFS and RDM operate at similar CPU efficiency levels for random workloads at smaller I/O block sizes. For sequential workloads, RDM shows improved CPU efficiency.

The performance tests described in this study show that VMFS and RDM provide similar I/O throughput for most of the workloads we tested. Most enterprise applications, such as Web server, file server, databases, ERP, and CRM, exhibit I/O characteristics similar to the workloads we used in our tests and hence can use either VMFS or RDM for configuring virtual disks when run in a virtual machine. However, there are 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

- Processors: 2 dual-core Intel Xeon processor 5160, 3.00GHz, 4MB L2 cache (4 cores total)
- Memory: 8GB
- Local disks:
  1 Seagate 146GB 10K RPM SCSI (for ESX Server and the guest operating system)
  5 Seagate 146GB 10K RPM SCSI in RAID 0 configuration (for the test disk)

Software

- Virtualization software: ESX Server 3.0.1 (build 32039)

Guest Operating System Configuration

- Operating system: Windows Server 2003 R2 Enterprise Edition 32-bit, Service Pack 2, 512MB of RAM, 1 CPU
- Test disk: 10GB unformatted disk

Iometer Configuration

- Number of outstanding I/Os: 8
- 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

- 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 the chapter "Using the esxtop Utility" in the VMware Infrastructure 3 *Resource Management Guide* at
  http://www.vmware.com/pdf/vi3_301_201_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_301_201_server_config.pdf
- VMware Infrastructure 3 *Server Configuration Guide*
  http://www.vmware.com/pdf/vi3_301_201_server_config.pdf

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

Revision 20071207 Item: PS-035-PRD-01-01