



Steve Garraway  
Jon Tate

# VMware Proof of Practice and Performance Guidelines on the SAN Volume Controller

## Overview

This IBM® Redpapers™ publication presents the results of I/O performance tests conducted using VMware ESX running on IBM System x® servers attached to the IBM System Storage™ SAN Volume Controller (SVC). We show the results of tests involving different workloads and I/O patterns and offer guidance about how best to use the SVC with VMware. We reference the SVC performance best practices that are detailed in various IBM Redbooks® publications, as well as VMware documents.

## Introduction

In this IBM Redpapers publication, we discuss performance configuration guidelines that can be used when utilizing SVC for use with VMware ESX. We attempt to answer some basic configuration questions system administrators and storage administrators may have when utilizing an SVC in ESX environments.

The I/O generator used during the performance measurement phase is *Iometer*. It is both a workload generator and a measurement tool, and some performance measurements were taken with Iometer. A single instance of Iometer was run on each ESX virtual machine (VM). Repeated runs with different I/O characteristics (Random I/O, Sequential Reads and Writes, and Random Reads and Writes) were run against each configuration.

This paper is not meant as a statement of maximum possible I/O benchmark numbers for SVC. Those numbers are stated in the Storage Performance Council SPC-1 and SPC-2 results posted by IBM on the SPC web site, found at:

<http://www.storageperformance.org/results>

We ran controlled experiments on a given set of hardware where only one variable was changed at a time, that is, one ESX server with 12 VMs, two ESX servers with 6 VMs, and three ESX servers with four VMs. We drew conclusions based on the effect that changing one configuration item had on performance.

All of the performance tests referred to in this paper were conducted at the Open Systems Lab based at IBM Almaden in San Jose, California.

## Assumptions

We start with the assumption that the best practices that have been documented by the SAN Volume Controller performance architects are relevant for a wide range of applications, including VMware ESX server farms.

We also assume that the use of Iometer for workload generation is appropriate for determining storage performance for VMware ESX environments.

We also assume that the reader is familiar with SAN fundamentals, as well as ESX, SVC, and IBM System Storage DS8000® basics. Therefore, there will be no detailed discussion of these topics.

Lastly, it is not our intention to document the best practices of SVC configuration here. Instead, we refer you to the relevant IBM Redbooks publications for these details. In this paper, we concentrate on how best to configure SVC VDisks with VMware ESX.

## Intended audience

The intended audience of this paper is any technical lead, system administrator, or storage administrator that works in a production environment that is part of an effort to implement the IBM System Storage SAN Volume Controller as storage for VMware ESX servers. We give the technical staff a place to start, based on tested and documented best practices, when designing and implementing their configurations.

## Objective

The objective of these tests is to take a typical VMware environment (in this case, three rack servers running several VMs each) and generate as much I/O as possible to see how various VDisk configurations affect performance. From these results, we draw conclusions in order to make recommendations.

Secondly, we observe the impact of performing VMware metadata operations, such as virtual disk create/delete, and make recommendation to minimize that impact.

## The test environment

We obtained the performance figures presented in this paper using the following hardware.

### **SAN Volume Controller**

We created two I/O groups using 2145 model 8G4 nodes. These nodes have 4 Gbps HBAs and were running code level 4.3.1.3 (build 9.14.0901210200).

In our testing, we used a single fabric solution. A fully redundant SAN should be used for production SVC environments, which is typically achieved by using a dual fabric solution.

### **IBM System Storage DS8000**

We used a DS8300 model 932 with 64 GB cache and 2 GB NVS to provide the RAID storage for the SVC nodes. We allocated a total of eight arrays using 300 GB 15K DDMs and RAID 5. We automatically reserved hot spare DDMs for the DS8000 (as normal). We used two ports on each of the two 4 Gbps host adapters for SAN connectivity, with four connections in total. The image was running Release 3 code, specifically LMC 5.3.1.116.

### **IBM System Storage SAN b-type switches**

The DS8000 and SVC nodes were connected directly to blades in an IBM System Storage SAN768B (2499-384) director using 8 Gbps capable ports. The director was running FOS 6.2.0e.

The host servers were connected to two IBM System Storage SAN32B-3 (2005-B5K) switches using 4 Gbps capable ports, which then connected to the director by way of sufficient ISLs to avoid contention. The switches were running FOS 5.3.1.

### **IBM System x servers**

The VMware hosts were comprised of one IBM eServer™ xSeries® 445 (x445) server and two xSeries 440 (x440) servers, each with 16 GB of RAM. The x445 had four Intel® Xeon 3.0 GHz processors, while the x440s each had four Intel Xeon® 2.4 GHz processors.

The servers were each SAN attached using two QLE2340 2 Gbps HBAs, with firmware 3.03.19 and driver 7.08-vm33.3. Each HBA was zoned to see one port on each SVC node in an I/O group, giving a total of four host paths per I/O group.

The level of VMware ESX installed was 3.5 U3 (build 123630) and VMFS was at Version 3.31 with a block size of 1 MB.

## Summary

Figure 1 gives a high level overview of the design.

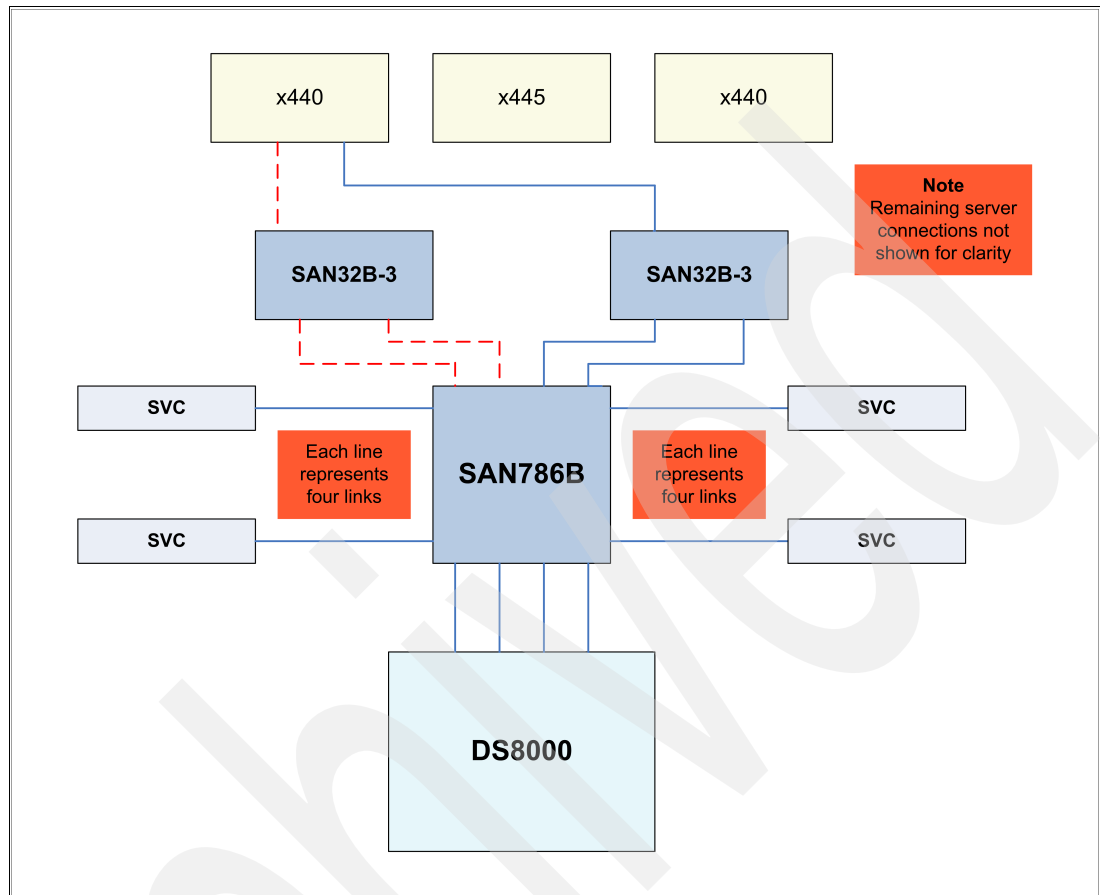


Figure 1 Test environment overview

## Configuration

As previously stated, you should refer to the appropriate IBM Redbooks publications for full details about configuring the various components. This section only provides a summary.

## Best practices

It is important to follow best practices in the configuration of all components of the solution, particularly for production environments, as this will help ensure optimal performance and availability. The following areas should be configured according to the relevant product manuals and IBM Redbooks publications:

- ▶ SAN zoning of:
  - SVC
  - DS8000
  - ESX hosts

- ▶ DS8000:
  - RAID arrays and ranks
  - Extent pools
  - Balance ranks across both server complexes
  - Number of host adapter ports
- ▶ SVC:
  - MDisks
  - MDisk Groups
  - Vdisks
- ▶ Host configuration:
  - Number of paths per HBA to SVC
  - The correct multipathing options

### **IBM Redbooks publication references**

SAN switch configuration and details of aliases and zoning are covered in *Implementing an IBM/Brocade SAN with 8 Gbps Directors and Switches*, SG24-6116.

For information about the DS8000 and how to configure it, refer to *IBM System Storage DS8000: Architecture and Implementation*, SG24-6786.

For guidance about the optimal configuration of DS8000 LUNs for use by the SVC, refer to *DS8000 Performance Monitoring and Tuning*, SG24-7146.

*SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521 is the best source for optimal configuration of the SVC and its back-end storage. It also contains a chapter about host connectivity.

For more general information about the SVC, refer to *Implementing the IBM System Storage SAN Volume Controller V4.3*, SG24-6423.

## **Logical configuration steps**

Assuming all of the hardware has been installed, cabled, and powered on, the following logical steps will configure the different components ready for testing.

### **SAN**

Zone the various components according to the best practices in “Best practices” on page 4. Remember to limit the maximum number of paths from each ESX to four per VDisk.

### **DS8000**

Each of the eight array sites will provide one RAID 5 array, which in turn is used to define one fixed block (FB) rank in its own extent pool. This will allow to control LUN creation on specific RAID arrays.

Perform these steps:

1. Configure the host ports as SCSI FCP.
2. Configure the eight RAID 5 arrays from the eight array sites.
3. Create the eight fixed block (FB) extent pools.
4. Configure the eight fixed block ranks on the RAID arrays and assign them to the extent pools.
5. Create eight volume groups of type *scsimask*.
6. Create the LUNs and assign them to the appropriate extent pool and volume group.

7. Create the host connections for the volume groups (remember that every SVC node must be able to see all LUNs assigned to the SVC cluster).

## SVC

Perform these steps:

1. Discover the MDisks.
2. Assign meaningful names for the discovered storage controllers.
3. Assign meaningful names to the discovered MDisks.
4. Create MDisk Groups and assign MDisks to the groups.
5. Define the VMware hosts.
6. Create the VDIs.
7. Map them to the hosts.

## Configure VMware host multipathing

Once the VDIs have been created and the host zoning is in place, the ESX host must discover the VDIs. This can be done from either the ESX CLI (run `esxcfg-rescan vmhbaX`, where X is the hba number) or from the VMware Infrastructure Client (select **Configuration** → **Storage Adapters** → **rescan**). Then create VMFS volumes from the ESX CLI or from the VMware Infrastructure Client (select **Configuration** → **Storage** → **Add Storage**).

To ensure that the zoning is correct, verify that the VDIs are visible on all four paths. This can be done from either the ESX CLI (run `esxcfg-mpath -1`) or from the VMware Infrastructure Client (select **Configuration** → **Storage** and select each individually added storage disk).

The default multipathing policy for ESX V3.5 with SVC is Most Recently Used (MRU). Using MRU is not recommended in an SVC environment for the following reasons:

1. SVC assigns a preferred node for each VDI within the I/O group. Prior to SVC V4.3.1.5, sending I/O to the non-preferred node incurred a performance penalty, which could be avoided by configuring the correct path to the preferred node. Running SVC V4.3.1.5 or later avoids the need for this additional configuration step.
2. MRU will eventually end up sending all I/O through a single path/node. This seems to happen even during normal operation with no paths going down.

To ensure that the workload is evenly distributed, the VDIs need to be created with their preferred nodes evenly spread. This is the default if VDIs are created without specifying the preferred node. Once the VDIs have been created, display the preferred SVC nodes using the following command:

```
SVC14:admin> svcinfo lsvdisk ESX_DG_DATA1
```

Within ESX, the path to the preferred node can be specified from the VMware Infrastructure Client. Select **Configuration** → **Storage** and select a storage volume, right-click it, and select **Properties**. Select the **Fixed Multipathing** policy and then select the correct path to the SVC preferred node.

**Important:** When using ESX V3.5, there must be no I/O going to the selected storage volume when this operation is done or it will not work.

# Performance monitoring tools

These are some of the tools we used in creating this IBM Redpapers publication.

## IBM Tivoli Storage Productivity Center for Disk Version 3.3.2

Tivoli® Storage Productivity Center for Disk is an essential tool in an IBM SAN environment to ensure the health of the data center. Tivoli Storage Productivity Center for Disk offers the following features:

1. A graphical overview of the entire data center topology, from hosts to Fibre Channel switches to storage.
2. The ability to drill down into each object in the topology. For example, you can select a given DS8000 and expand it to view all of the layers of the virtualization hierarchy.
3. Collection of very detailed performance data on LUNs, RAID arrays, switches, and so on. For example, for a given LUN over a specified time period, you can see the IOPS, the response time, the throughput, and the read or write cache hit ratio.
4. A wide range of reports that can be used to analyze the collected performance data.

Tivoli Storage Productivity Center for Disk is a great way to monitor and report on all of the layers of the technology stack in the IBM SAN environment and is a critical component of the setup in a data center.

**Attention:** Although Tivoli Storage Productivity Center for Disk V3.3.2 was used while collecting the data for this paper, the current release at the time of writing is IBM Tivoli Storage Productivity Center for Disk Version 4.1, which is covered in *IBM Tivoli Storage Productivity Center V4.1 Release Guide, SG24-7725*.

## VMware performance monitoring utility: esxtop

The *esxtop* command-line utility provides a detailed look at how ESX Server uses resources in real time. It can be started in one of three modes:

- ▶ Interactive (the default)
- ▶ Batch
- ▶ Replay

*esxtop* was run in batch mode from each of the three ESX servers, resulting in a file with a .csv (comma-separated values) extension. Data was taken every 300 seconds so that the data interval would be the same as the Tivoli Storage Productivity Center for Disk data.

There is no way to select the data that you want output in *esxtop* batch mode. Because the .csv file created by *esxtop* has more columns than can be read into Microsoft® Excel 2003 (4K maximum) or 2007 (16K maximum), the data had to be reduced to only the desired columns. Windows® Perfmon was used for this purpose. It is a slow and manual process, but resulted in a .csv file that was able to be imported into the Excel spreadsheets used for the Tivoli Storage Productivity Center for Disk data.

There are many pages on the VMware Web site that cover the usage of esxstop. Two such pages are:

[http://pubs.vmware.com/vi3i\\_e35u2/resmgmt/wwhelp/wwhimpl/common/html/wwhelp.htm?context=resmgmt&file=vc\\_esxstop.14.1.html](http://pubs.vmware.com/vi3i_e35u2/resmgmt/wwhelp/wwhimpl/common/html/wwhelp.htm?context=resmgmt&file=vc_esxstop.14.1.html)

[http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=1008205](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1008205)

## Iometer workload generator

*Iometer* is an I/O subsystem measurement and characterization tool for single and clustered systems. Iometer is both a *workload generator* (that is, it performs I/O operations in order to stress the system) and a *measurement tool* (that is, it examines and records the performance of its I/O operations and their impact on the system).

Iometer consists of two programs, *Iometer* and *Dynamo*:

- ▶ Iometer is the controlling program. Using Iometer's graphical user interface, you configure the workload, set operating parameters, and start and stop tests. Iometer tells Dynamo what to do, collects the resulting data, and summarizes the results in output files. Only one copy of Iometer should be running at a time; it is typically run on the server machine.
- ▶ Dynamo is the workload generator. It has no user interface. At Iometer's command, Dynamo performs I/O operations and records performance information, then returns the data to Iometer. There can be more than one copy of Dynamo running at a time; typically, one copy runs on the server machine and one additional copy runs on each client machine. Dynamo is multithreaded; each copy can simulate the workload of multiple client programs. Each running copy of Dynamo is called a *manager*; each thread within a copy of Dynamo is called a *worker*.

Further information about Iometer can be obtained from the following Web site:

<http://www.iometer.org/>

## Testing methodology

The following section describes the individual Iometer test runs that will be considered.

4 KB block sizes and 64 KB block sizes were used for each scenario and were distributed 50/50, with varying percentage combinations of reads and writes. The combinations tested were:

- ▶ 70% reads and 30% writes
- ▶ 100% sequential reads
- ▶ 100% random reads
- ▶ 100% sequential writes
- ▶ 100% random writes

**Note:** The combination of 70% reads and 30% writes is often cited as fairly representative of a typical database workload.



## Storage configuration variations

Here are the storage configurations that will be considered for the DS8000, SVC, and ESX.

### DS8000 configurations

Using SVC, VDisk striping across back-end storage RAID arrays can be accomplished by creating MDisk Groups, which consist of multiple MDisks, where each MDisk is mapped to a different back-end storage RAID array. A storage configuration with one LUN using all of the space on the RAID array is the SVC best practice.

Even and odd extent pools should be used, which will split the load across both server complexes in the DS8000 (in this case, two extent pools per processor complex).

Despite this best practice recommendation, it was not possible for each LUN to occupy the entire single-rank extent pool due to capacity allocated for other testing. However, there was no other significant I/O activity on these shared extent pools during these tests.

#### **Configuration 1**

The DS8000 will be configured to connect to the SVC and will present to the SVC eight 100 GB LUNs, one from each of the eight single-rank extent pools/RAID arrays. These LUNs will be visible as MDisks to the SVC, and placed in a single MDisk Group. VDIsks will then be created from these MDisks and presented to front-end ESX hosts as LUNs. These will be used in the first scenario described in “Configuration 1” on page 10.

#### **Configuration 2**

The second scenario, described in “Configuration 2” on page 11, required the creation of eight more LUNs on each of the single-rank extent pools. However, due to the previously mentioned limited space availability, 80 GB LUNs were created on six of the RAID arrays, while only 10 GB was available on each of the other two. These eight additional LUNs were added to the existing MDisk Group on the SVC before the single large VDisk could be created.

### SAN Volume Controller configurations

The basic setup of the SVC followed most of the best practice guidelines. The lab environment available for this paper did not allow the use of a dual-fabric SAN topology, so to come as close as possible, the storage connections (both DS8000 and SVC) were split across two separate blades of the DCX director class switch. The host connections were made to different physical switches that were ISL connected to the DCX. This configuration is shown in Figure 1 on page 4.

Recommendations for zoning aliases and zones, and the proper configuration of MDisks, MDisk Groups, VDIsks, and Hosts were all followed.

Here are specific details regarding some of the SVC configuration variations:

1. Benchmarks were run using an MDisk Group extent size of 256 MB.
2. Each DS8000 LUN presented to the SVC was from a single-rank extent pool, but did not consume all of the space on that extent pool.

3. VDisk sizes will be of various sizes as required for the specific ESX configurations being set up. All VDIs will be striped across all the MDIs in the MDI Group.
  - a. The first scenario is to create a single VDI for each ESX VM being created, plus four I/O VDIs. This will allow for distributing the VDI access across SVC nodes.
  - b. The second scenario is to create a single large VDI, which will restrict access to one SVC node. ESX will then create the necessary virtual disks needed for each VM as files within the VMFS.
4. Benchmarks will be run where VDIs are using multiple VDIs on two I/O groups (the first scenario above) and then again where the single large VDI uses a single I/O group (the second scenario above).

### **ESX VM configurations**

There are two configurations used in this benchmark. The configuration normally recommended by VMware is the second one, as it simplifies the disk LUN requirements and administration.

#### ***Configuration 1***

The first configuration uses an individual 8 GB file in a VMFS on a 20 GB VDI for each VM, into which Windows Server 2003 is installed. This results in every Windows /C: drive being in its own VMFS in its own VDI.

Additionally, there are four separate 50 GB VDIs utilized for all I/O and shared by all VMs. Because all ESX hosts in the complex were defined within a single SVC host object, all shared I/O VDIs have the same SCSI ID and are able to be shared with no further action required.

A 50 GB VMFS is created on the VDI, and one 1.6 GB virtual disk (vmdk file) is created in it for each VM. This is repeated for each of the other three I/O VDIs. The 1.6 GB disks are then added to each VM, discovered by Windows, and left uninitialized for raw I/O. Each Windows VM now has four 1.6 GB I/O disks for its use by Iometer.

Figure 2 shows the MDisk Group, which consists of the eight 100 GB MDisks with the four 50 GB VDisks and two of the multiple 20 GB Windows VDisks. The single 8 GB and four 1.6 GB files are created for each VM.

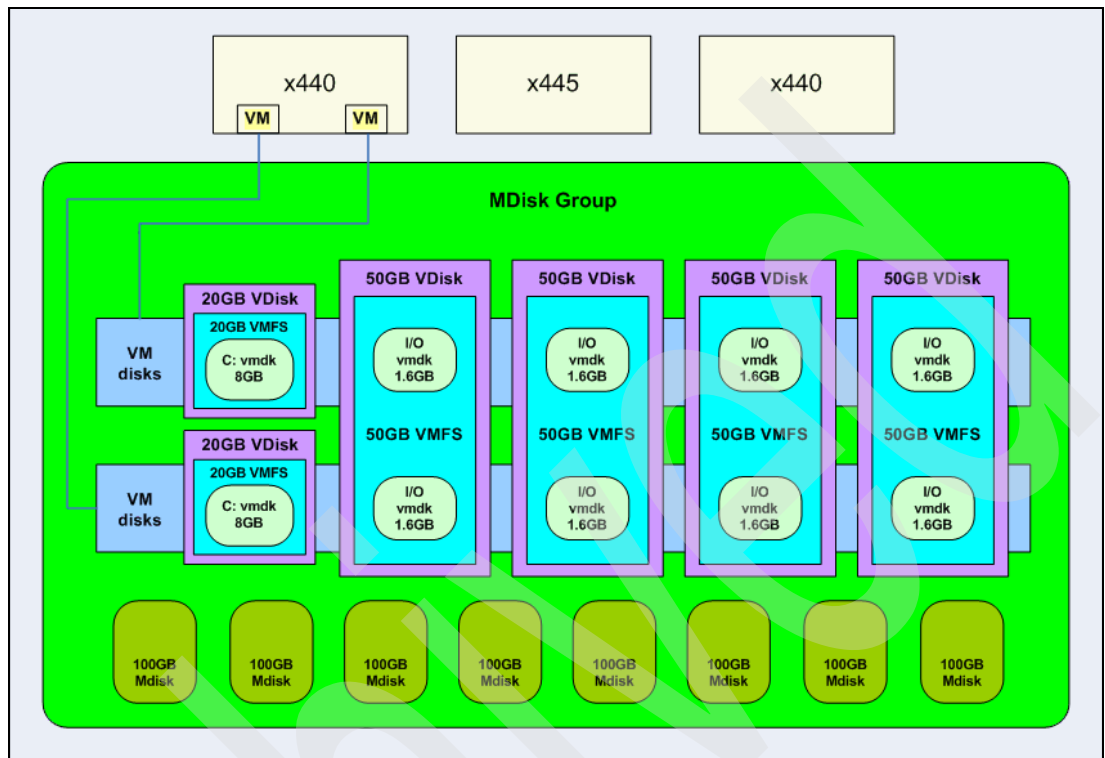


Figure 2 Individual VDisks per Windows installation with four shared I/O VDisks

VDisk I/O is distributed evenly across the four SVC nodes by configuring the ESX multipathing policy for each VDisk, as described in “Configure VMware host multipathing” on page 6.

### Configuration 2

The second configuration uses one VMFS on a single 800 GB VDisk. We define virtual disks for each Windows installation within it, plus the sets of four virtual disks (vmdks) needed by each VM for I/O with lometer.

While this option is documented by VMware and is often used because it simplifies the creation and management of storage disks in a large ESX server farm, it will result in potentially reduced performance, because all I/O will be forced through a single SVC node for the 800 GB VDisk. The benefit of distributing I/O across all available paths and utilizing all nodes of the SVC cluster will be gone.

Figure 3 shows the extra MDisks defined and the new 800 GB VDisk assigned to the three ESX hosts. One set of files are shown for each of two VMs; this setup is duplicated for all of the other VMs.

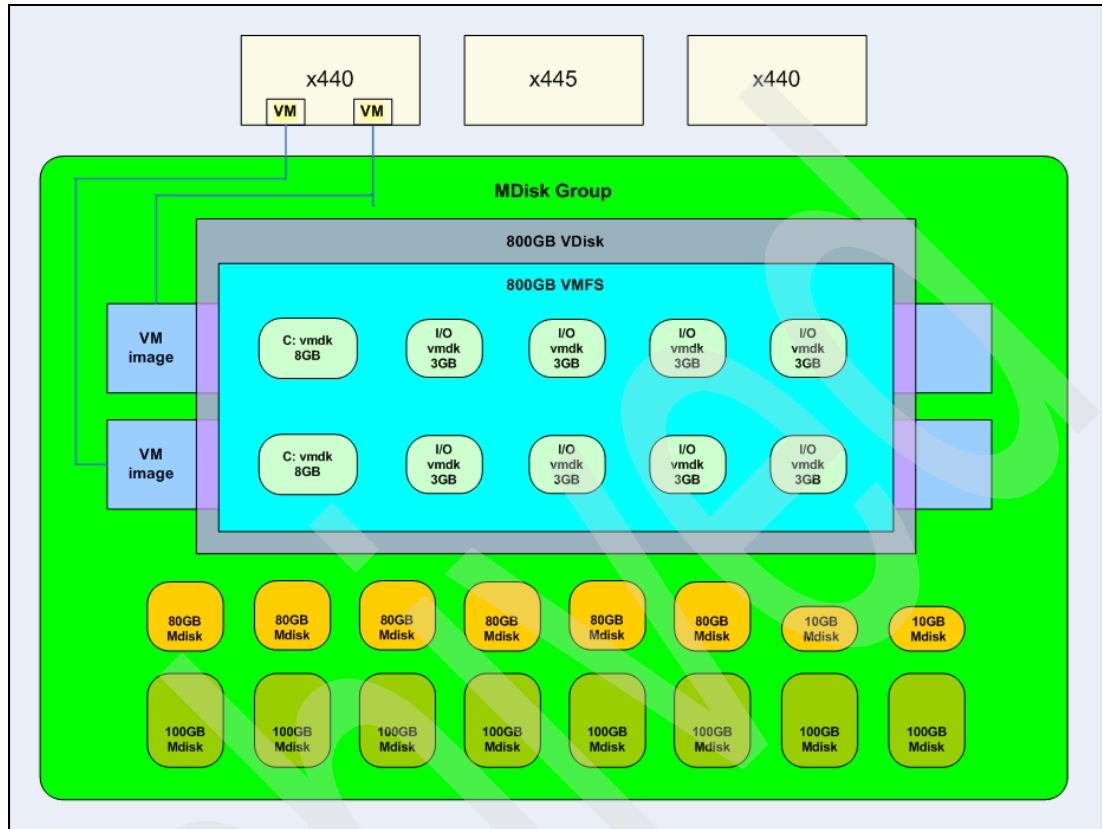


Figure 3 All VM disks in one large VDisk

## Test results

As mentioned in “Testing methodology” on page 8, 4 KB block sizes and 64 KB block sizes were used for each scenario and were distributed 50/50. The set of test cases was run five times with varying read/write and random/sequential characteristics.

## Scenarios

The I/O performance tests are grouped into two sets, each containing the five variants of R/W and random/sequential.

A third set of sets were conducted with a different aim, namely, trying to determine the impact to I/O of performing VMware metadata operations on LUNs. Such operations include:

- ▶ Expanding a VMFS datastore onto additional extents
- ▶ Powering on or off a virtual machine
- ▶ Acquiring a lock on a file
- ▶ Creating or deleting a file
- ▶ Deploying a virtual machine from a template
- ▶ Creating a new virtual machine
- ▶ Migrating a virtual machine with VMotion

## Test group 1

The following combinations of ESX hosts and VMs were tested:

- ▶ One ESX host with 12 VMs
- ▶ Two ESX hosts with six VMs each
- ▶ Three ESX hosts with four VMs each
- ▶ Three ESX hosts with six VMs each
- ▶ Three ESX hosts with eight VMs each

The first three cases all have a total of 12 VMs, but running on a different number of ESX hosts, while the last two cases keep the number of ESX hosts constant while increasing the number of VMs.

In all cases, the VMs were provisioned with disks as described in “Configuration 1” on page 10.

**Note:** All tests in test group 2 were also run with three ESX hosts with eight VMs each.

## Test group 2

For the second group of tests, the size of the I/O disks was increased from 1.6 GB to 3 GB.

For the first two tests, Windows was installed in its own VMFS/VDisk (as described in “Test group 1”). For the second two tests, all instance of Windows were installed in one VMFS in the 800 GB VDisk, as described in “Configuration 2” on page 11. Our goal is to show the performance benefits (if any) of spreading the load across multiple SVC nodes.

### Part A

The VMs were provisioned with disks as described in “Configuration 1” on page 10, except for the I/O disks being 3 GB instead of 1.6 GB. Two tests were run:

1. Each VM had four 3 GB I/O disks.
2. Each VM had one 3 GB I/O disk.

These two tests are designed to illustrate how restricting the number of I/O disks removes the ability to distribute each VMs workload across all available SVC nodes.

### Part B

All VMs were installed in a single large disk so they were sharing one VMFS (refer to “Configuration 2” on page 11).

The same two tests were run:

1. Each VM had four 3 GB I/O disks.
2. Each VM had one 3 GB I/O disk.

These two tests are designed to illustrate how sharing a single VDisk affects I/O performance.

## Test group 3

This group of tests have a different purpose than the first two, as they are intended to demonstrate the impact of metadata events on VM data I/O.

All tests were run with a single large VDisk, as described in “Configuration 2” on page 11. Two of the ESX hosts were used to generate I/O with eight VMs each running the 70/30 read/write test case, while the third ESX host issued the metadata commands.

The tests started with no metadata operations to set a baseline. Then a script was run to create and delete a 10 MB virtual disk on the VDisk, with a reducing interval between the events, using the following commands:

```
vmkfstools -c 10m path
vmkfstools -U path
```

## Results

The results are grouped in the same way as described in “Scenarios” on page 12.

### Test group 1

This group of tests illustrates the impact of different combinations of ESX hosts and VM count on the SVC.

#### SVC I/O rate

Figure 4 shows the I/O rate sustained by the SVC for the five workload types as the VM distribution is changed.

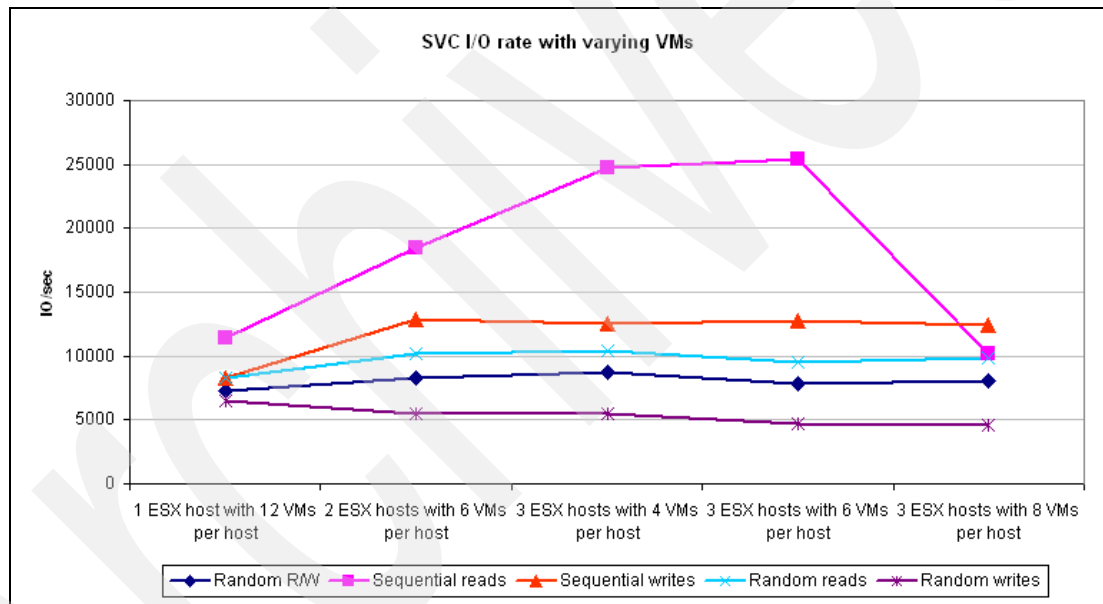


Figure 4 SVC I/O rate with varying VM distributions

For all workloads, this increases when going from one to two ESX hosts, but then remains almost constant except for the sequential read test case, which continues to increase with a third ESX host.

With most disk subsystems, read performance is typically better than write performance, both for sequential and random data access, due to large read caches and a lower RAID processing impact associated with reads than writes.

Sequential read access is considerably better than random read access for the following reasons:

- ▶ The SVC has “prefetch” logic, which attempts to read ahead from the back-end disk subsystem during sequential read scenarios.
- ▶ The DS8000 has an advanced caching algorithm called Sequential Adaptive Replacement Cache (SARC), which also performs prefetching when it detects sequential reads. For more details about SARC, refer to section 10.4.1, “Sequential Adaptive Replacement Cache” in *IBM System Storage DS8000: Architecture and Implementation*, SG24-6786.

Sequential read rates decrease when the number of VMs per ESX reaches eight (24 VMs in total). As we shall see later, the cache hit rate starts to decline at this point, so the additional six VMs are exceeding the rate at which the cache can pre-stage data from the MDisks.

### **DS8000 I/O rate**

Figure 5 shows the associated I/O rates in the DS8000 for the test cases.

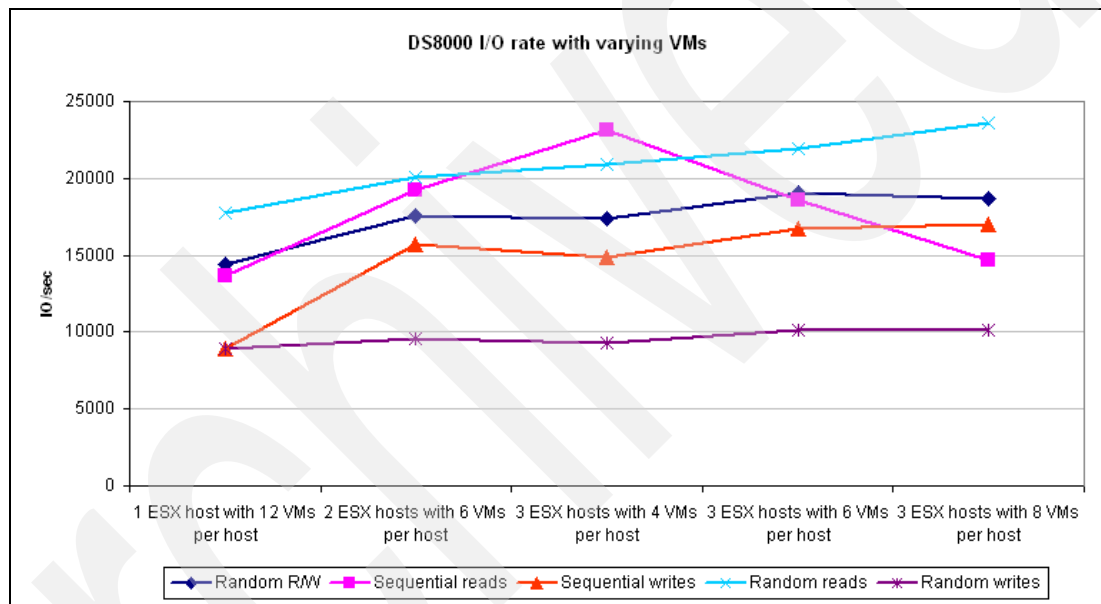


Figure 5 DS8000 I/O rate with varying VM distributions

The lines do not match exactly for various reasons, although the scale of the I/O rates is close to that of the SVC and the general trends are similar. Some of the reasons for the differences are:

- ▶ The SVC caches reads, so it may not always need to read data from the DS8000.
- ▶ The SVC caches writes and will coalesce writes where possible, hence potentially reducing the number of writes to the DS8000.
- ▶ Because the SVC stripes its VDisks across the MDisks, a host write may require updates to more than one DS8000 LUN.

### SVC response time

Figure 6 shows how the I/O response time varied during the test cases. It shows that response times start to increase as the number of VMs increases from 12 to 18 and then to 24.

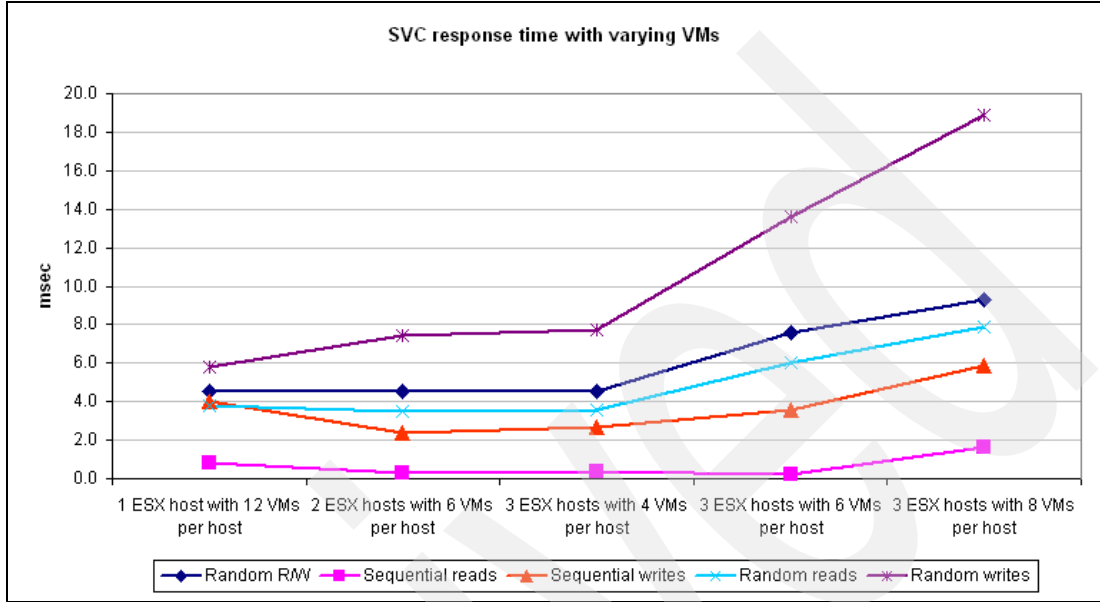


Figure 6 SVC response times with varying VM distributions

### DS8000 response time

Figure 7 shows the associated response times in the DS8000 for the test cases. For write requests, these are higher than the equivalent SVC figures, but because DS8000 writes do not contribute to SVC response times (provided the NVS in the DS8000 does not become full), this is not a problem.

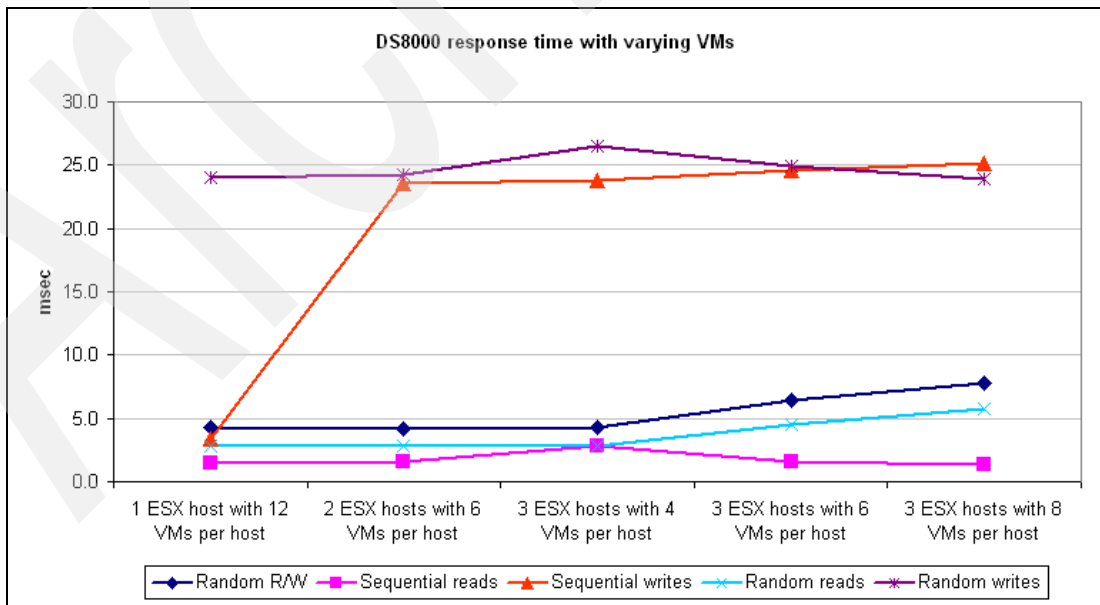


Figure 7 DS8000 response times with varying VM distributions



### SVC data rate

Figure 8 shows that the overall data rate through the SVC cluster remains almost constant regardless of the number of active VMs, except for the sequential read case. This initially grows almost linearly as the number of ESX hosts supporting the 12 VMs grows from one to two and then three.

The initial data point with 12 VMs on a single ESX host achieved 379 MBps, which is very close to the theoretical limit of two 2 Gbps HBAs. A second ESX host is required in order to exceed this data rate.

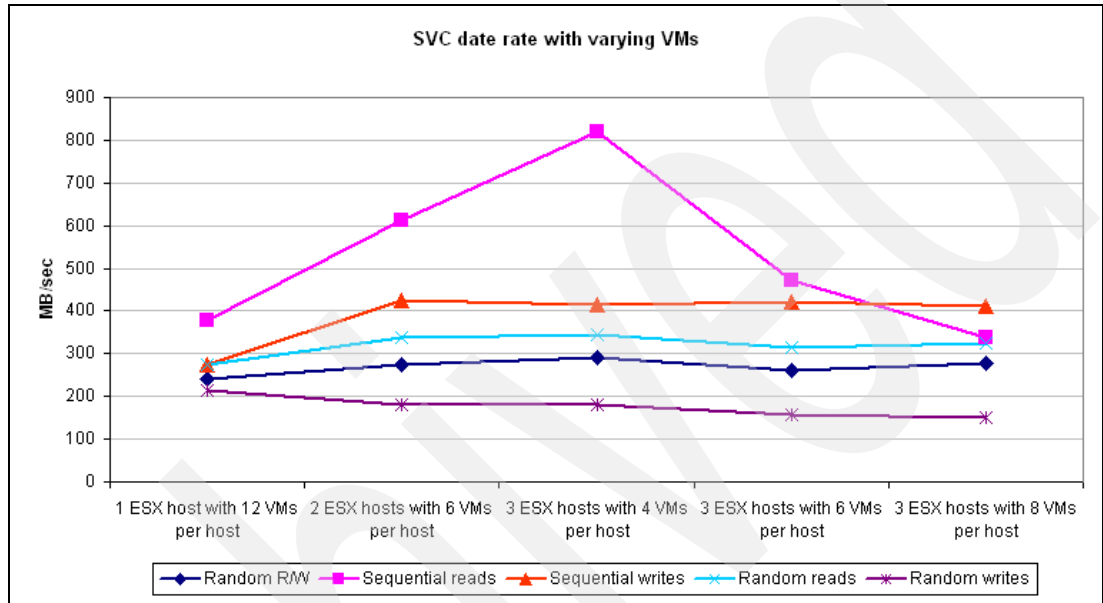


Figure 8 SVC date rates with varying VM distributions

Having peaked at the third data point, the sequential read performance starts to reduce as the number of VMs on each ESX host increases. This reflects the reduced cache hit rate shown in Figure 10 on page 19.

### DS8000 NVS delay

NVS delays occur when the DS8000 is unable to destage data fast enough to the physical disks. Figure 9 shows that this condition only occurred during the random write tests.

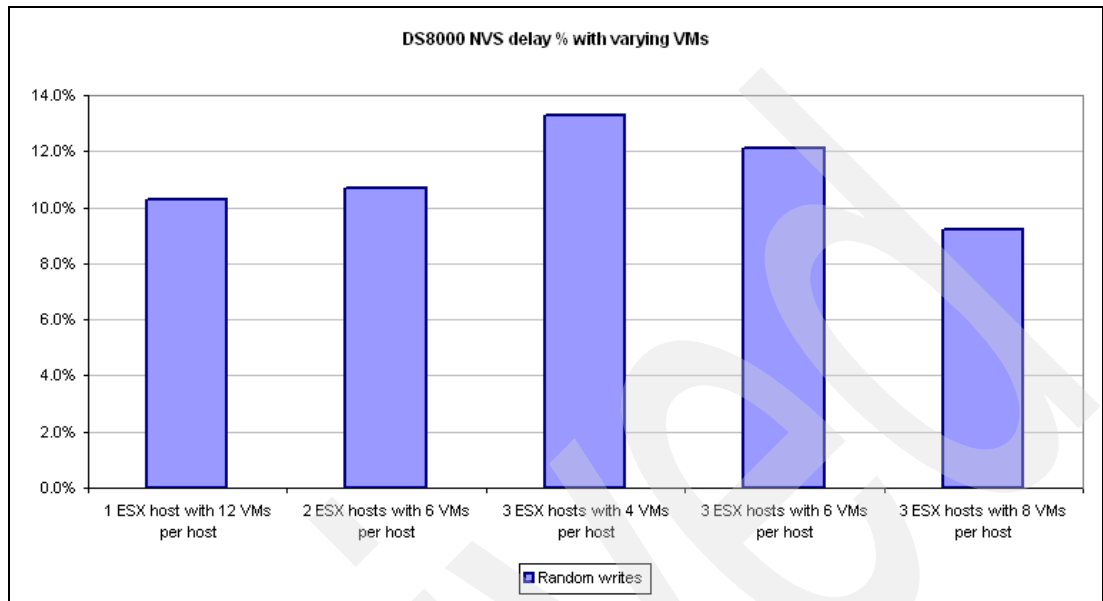


Figure 9 DS8000 NVS delays with varying VM distributions

Random writes to RAID 5 arrays are slower than reads because they require updating the parity information (the so called *write penalty*), which requires that existing data and parity be read first before the new data and new parity can be written. Sequential writes can often avoid this penalty if a complete stripe can be formed in cache before writing to disk, as there is then no need to perform the read operations first.

If the SVC detects that the performance of certain MDisks is degrading, it will attempt to limit the host data rate for the associated VDIs to try and prevent overloading of the back-end disk subsystem. This action effectively throttles the host I/O rate and makes the SVC performance appear worse. It is important to realize that this apparent SVC problem is actually a back-end disk problem.

### SVC cache hit rates

Figure 10 shows the benefit of read caching by the SVC. This benefit is limited to random reads, as the SVC cannot predict the blocks that will be requested, while for sequential reads the prefetch algorithm tries to have the requested data in the cache before the host ask for it. The near 100% figures for all sequential read tests show the effectiveness of this algorithm.

The cache hit rate starts to decline gradually when there are more than 12 VMs as the spread of I/O requests increases.

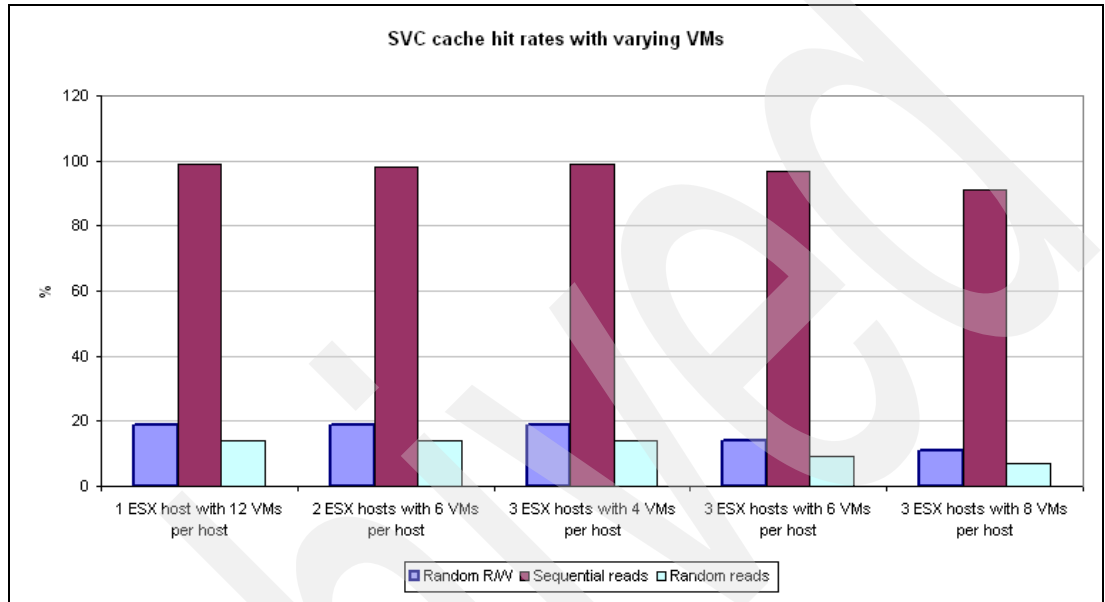


Figure 10 SVC cache hit rates with varying VM distributions

### **DS8000 cache hit rates**

Figure 11 provides the matching DS8000 cache hits. For sequential reads, the figures are very similar to those of the SVC for the same reasons. For the random reads, the DS8000 is achieving higher cache hit rates due to its larger cache size.

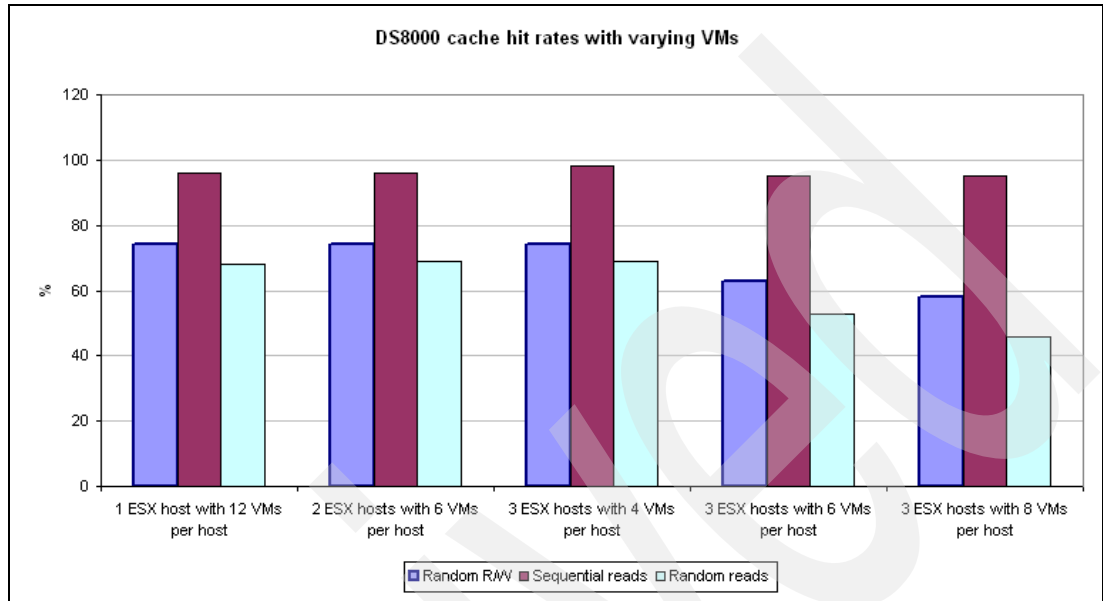


Figure 11 DS8000 cache hit rates with varying VM distributions

### **Conclusions**

We reach the following conclusions:

- ▶ The tests using only one ESX server were limited by the throughput of the 2 Gbps HBAs in the server.
- ▶ Sequential read performance was very good, up to the point where the MDisk throughput limit was reached.
- ▶ The random write test was the only case that showed non-zero NVS delays, which was caused by the limited number of MDisks (and hence RAID ranks) available in the DS8000.

### **Test group 2**

These four tests are intended to show the benefit of spreading the I/O load across four disks; as with correct pathing selection, this will allow each VM to utilize all four nodes in the SVC cluster.

The first graph in each pair shows four I/O vmdks with each on one of four 50 GB VDIs against one I/O vmdk on one 50 GB VDisk. The second graph of the pair shows four I/O vmdks against one I/O vmdk, but all sharing the one 800 GB VDisk.

### SVC I/O rate

Figure 12 shows how performance decreases for random data access with only one disk, while sequential access is almost unaffected.

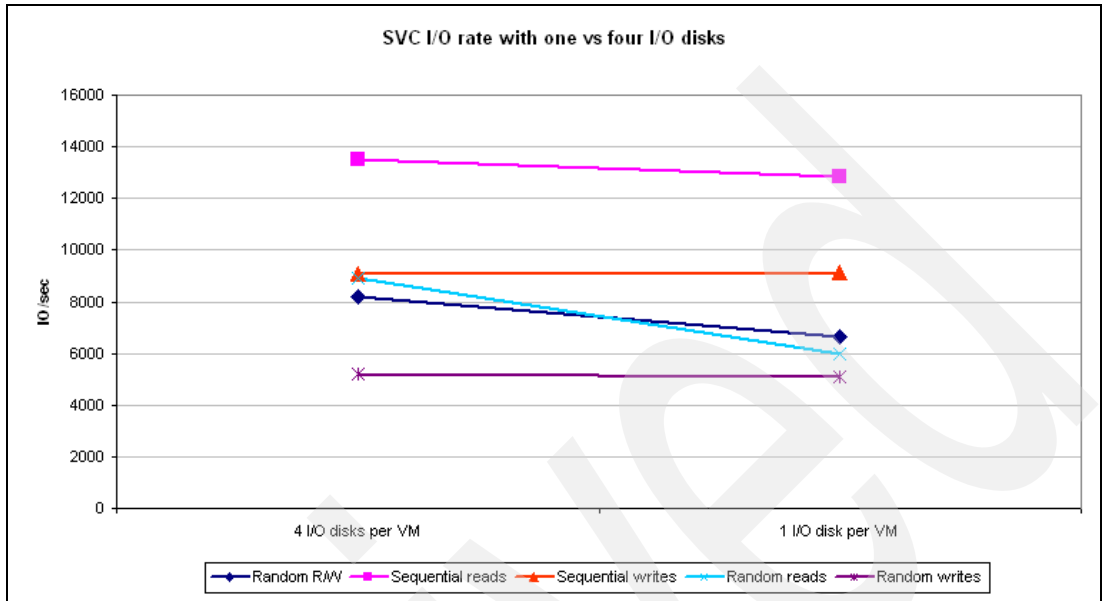


Figure 12 SVC I/O rate with four versus one I/O disks

Figure 13 shows the same tests, but with all VMs sharing one 800 GB VDisk. The results are almost identical, showing that in this scenario sharing a VDisk does not reduce performance.

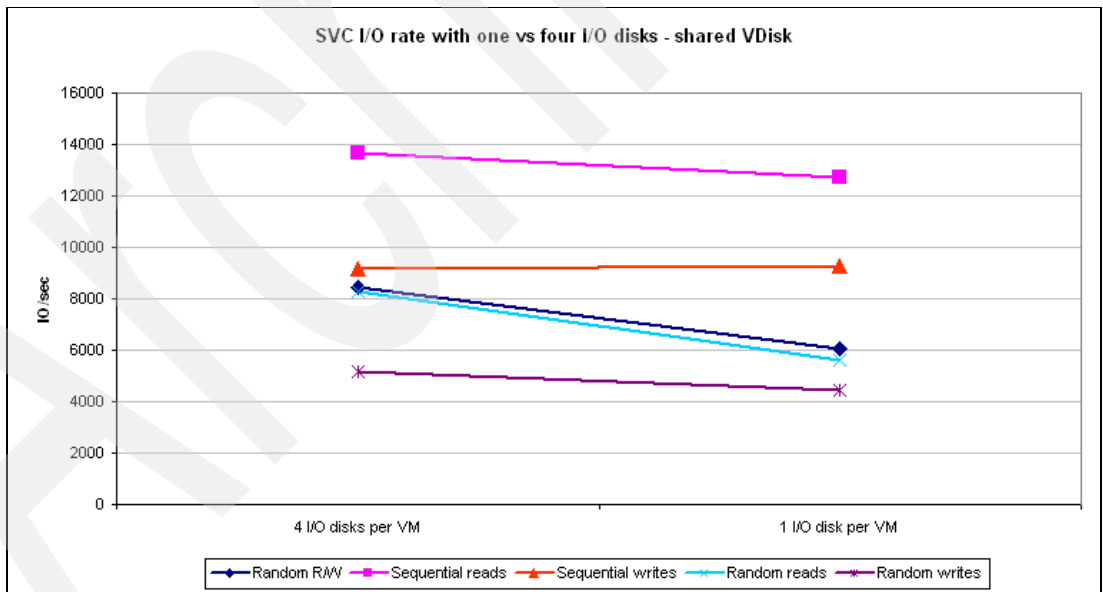


Figure 13 SVC I/O rate with four versus one I/O disks on a shared VDisk

**DS8000 I/O rate**

Figure 14 shows a similar reduction in I/O rates in the DS8000 when it is the SVC.

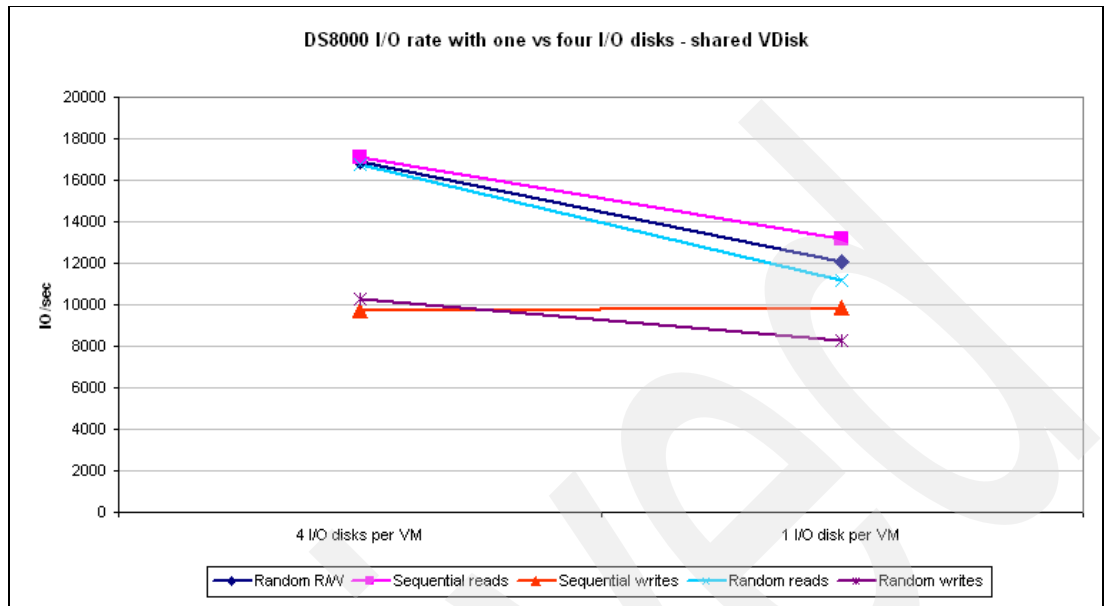


Figure 14 DS8000 I/O rate with four versus one I/O disks

Figure 15 are the equivalent tests with the 800 GB shared VDisk.

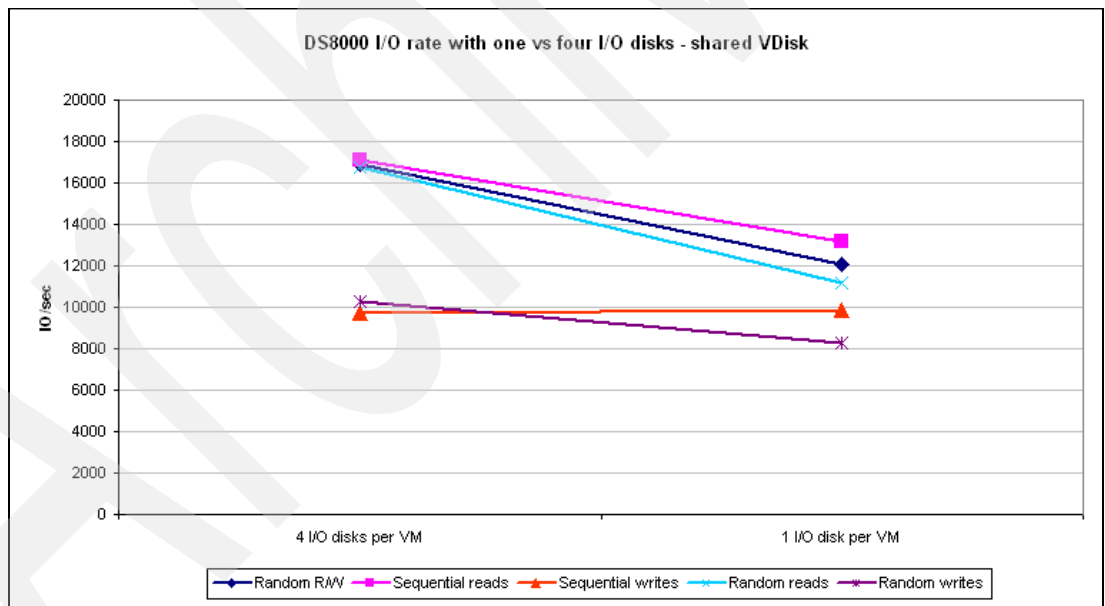


Figure 15 DS8000 I/O rate with four versus one I/O disks on a shared VDisk

### SVC response time

Figure 16 shows that response times are dramatically less with one disk than with four.

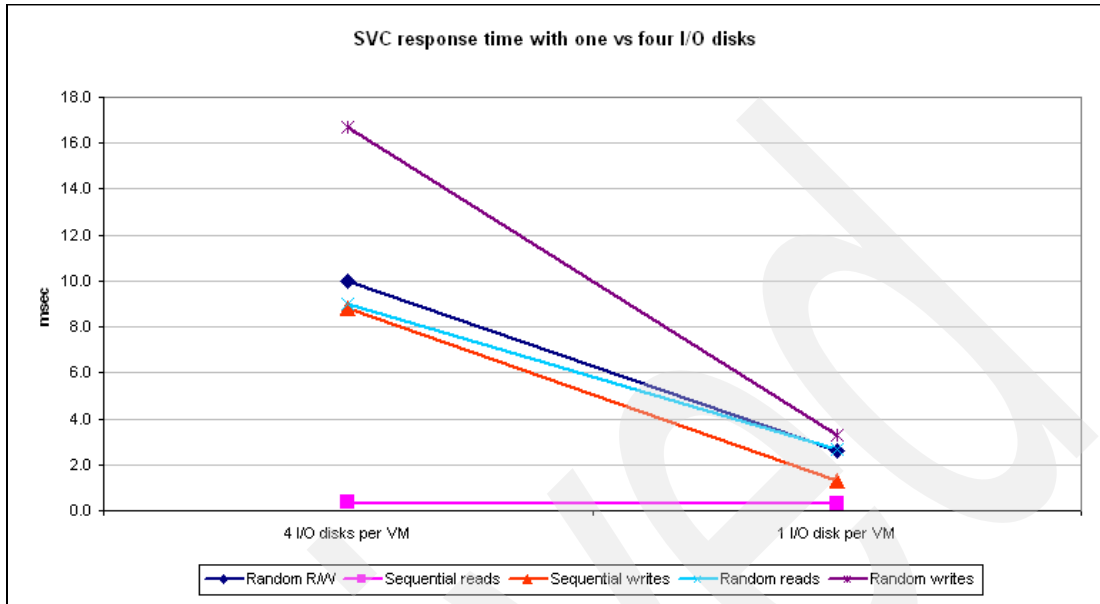


Figure 16 SVC response times with four versus one I/O disks

Figure 17 shows the equivalent tests with the 800 GB shared VDisk.

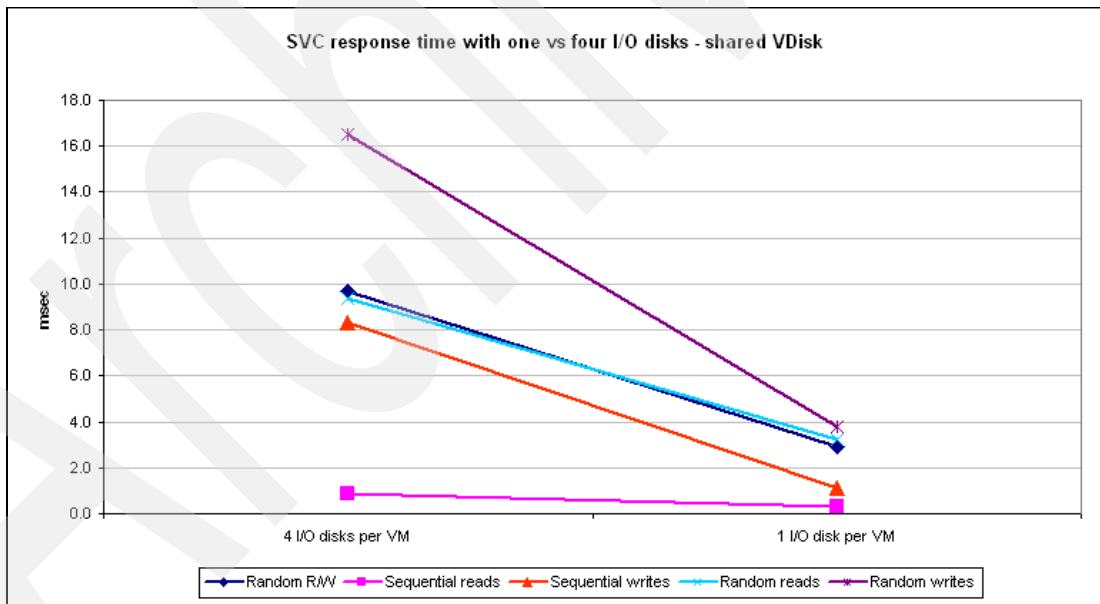


Figure 17 SVC response times with four versus one I/O disks on a shared VDisk

### DS8000 response time

Figure 18 also shows a reduction in DS8000 response times with one disk instead of four.

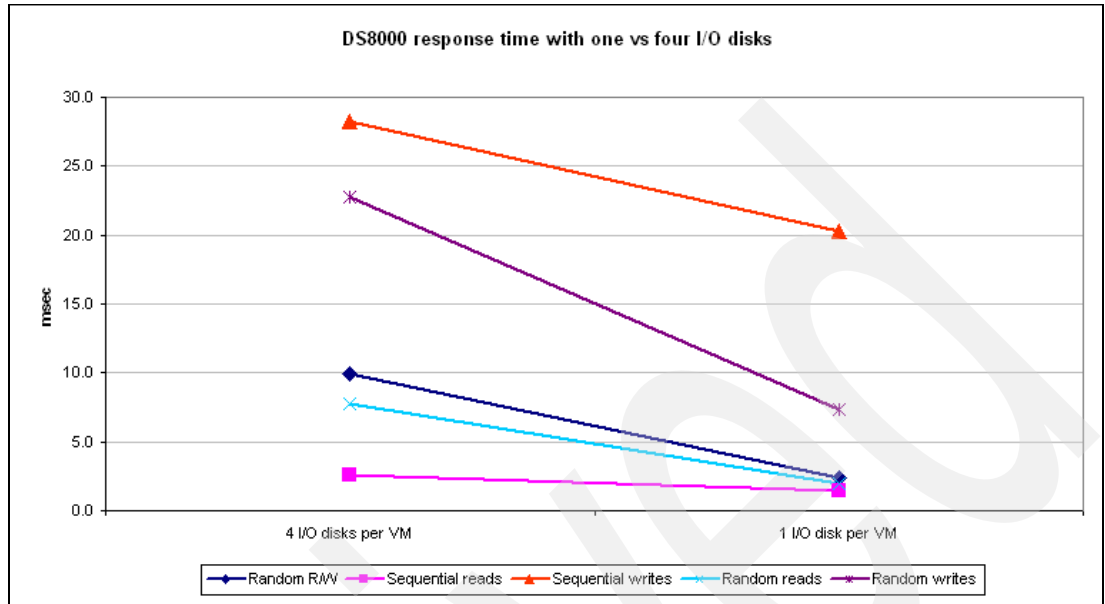


Figure 18 DS8000 response times with four versus one I/O disks

Figure 19 shows the equivalent tests with the 800 GB shared VDisk.

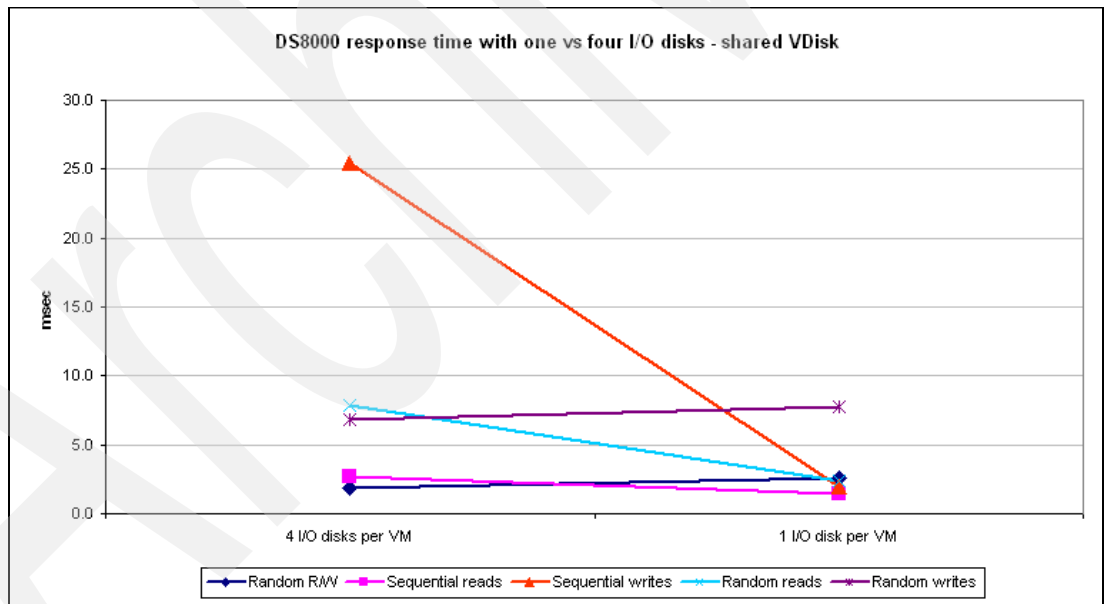


Figure 19 DS8000 response times with four versus one I/O disks on a shared VDisk



### SVC data rate

Figure 20 again shows the reduction in random read performance when all data is on only one disk, along with a lesser reduction in the read/write test.

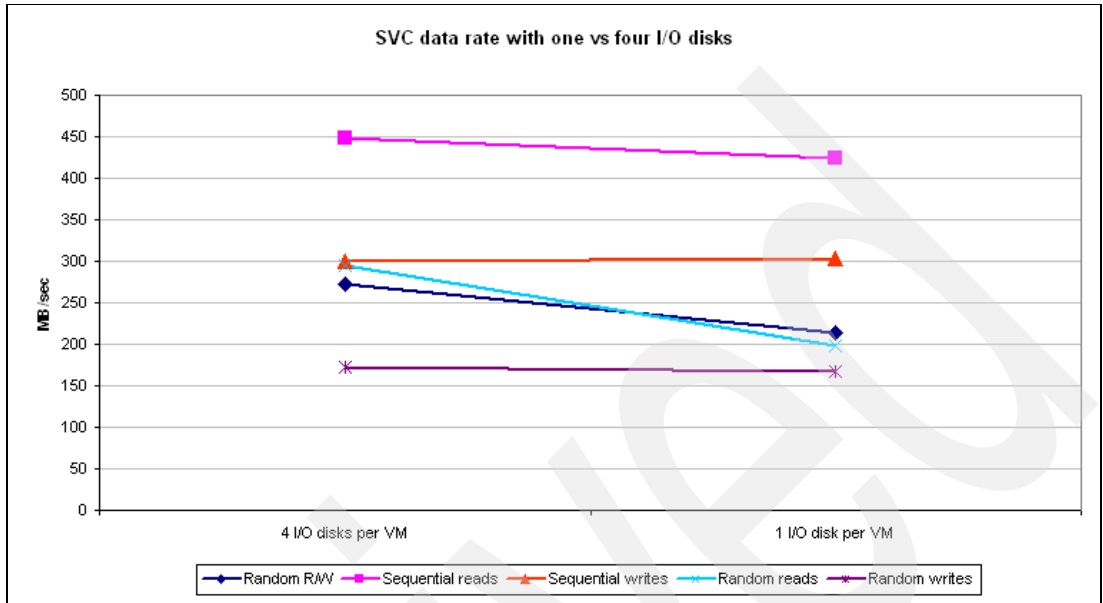


Figure 20 SVC data rates with four versus one I/O disks

Figure 21 shows the equivalent tests with the 800 GB shared VDisk.

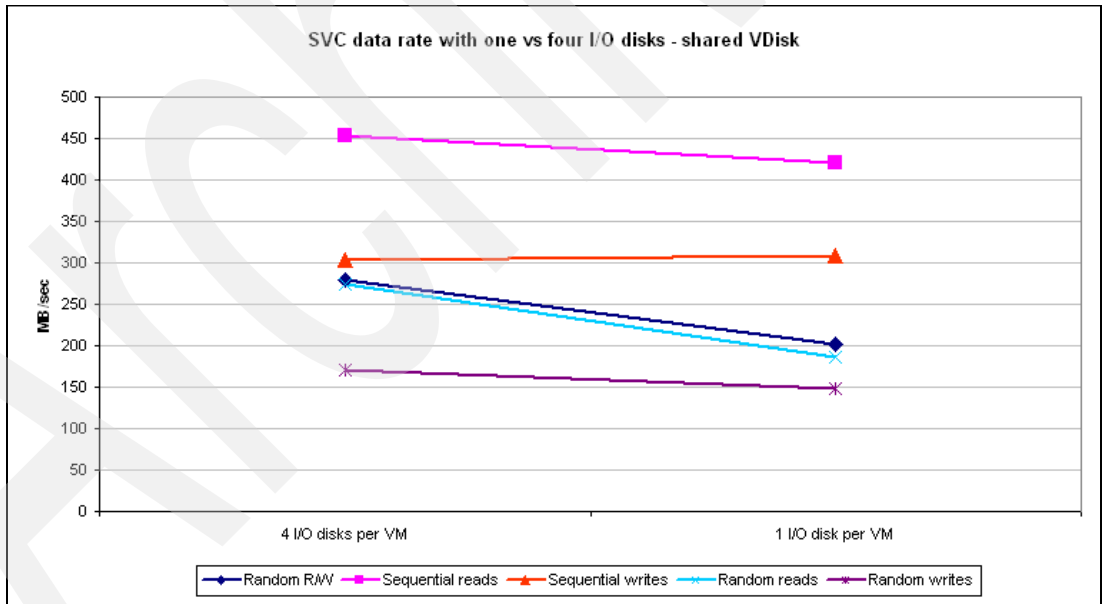


Figure 21 SVC data rates with four versus one I/O disks on a shared VDisk

### DS8000 NVS delay

Figure 22 only applies to the random write case, and shows that the NVS in the DS8000 was a bottleneck for this test case. In all other test cases, the NVS delay value was zero.

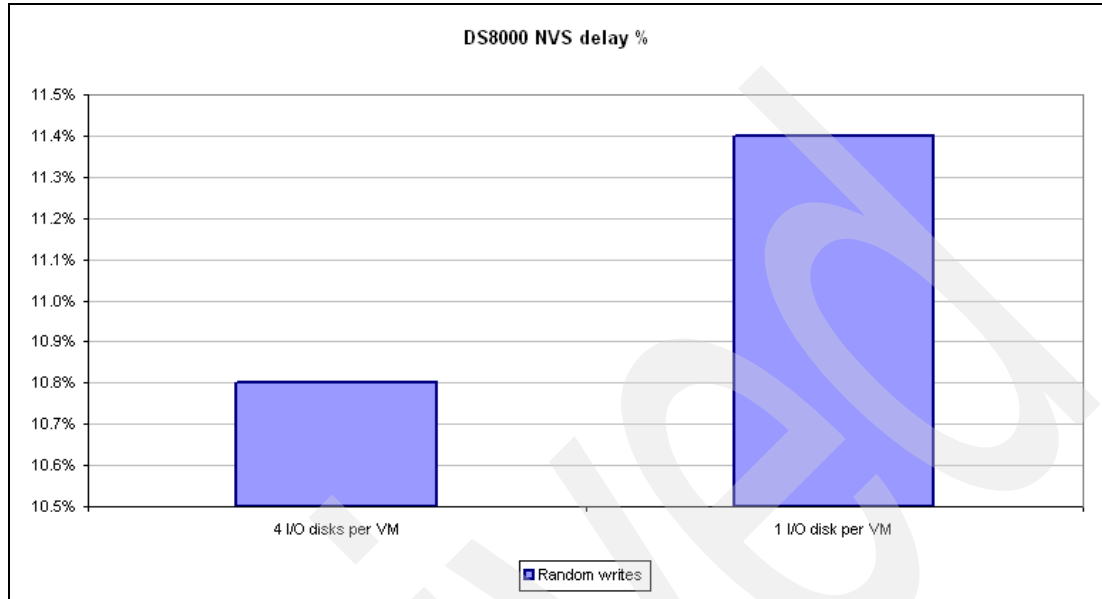


Figure 22 DS8000 NVS delays with four versus one I/O disks for random writes

This output explains the reduction in I/O rates for random writes in the DS8000 in Figure 14 on page 22.

If the SVC detects an increase in response times for MDisks, it will attempt to reduce the rate at which it destages data to the back-end disks to avoid overloading them. This action effectively throttles the host I/O rate, so it makes the SVC performance appear worse. It is important to realize that this apparent SVC problem is actually a back-end disk problem.

Figure 23 shows the equivalent tests with the 800 GB shared VDisk.

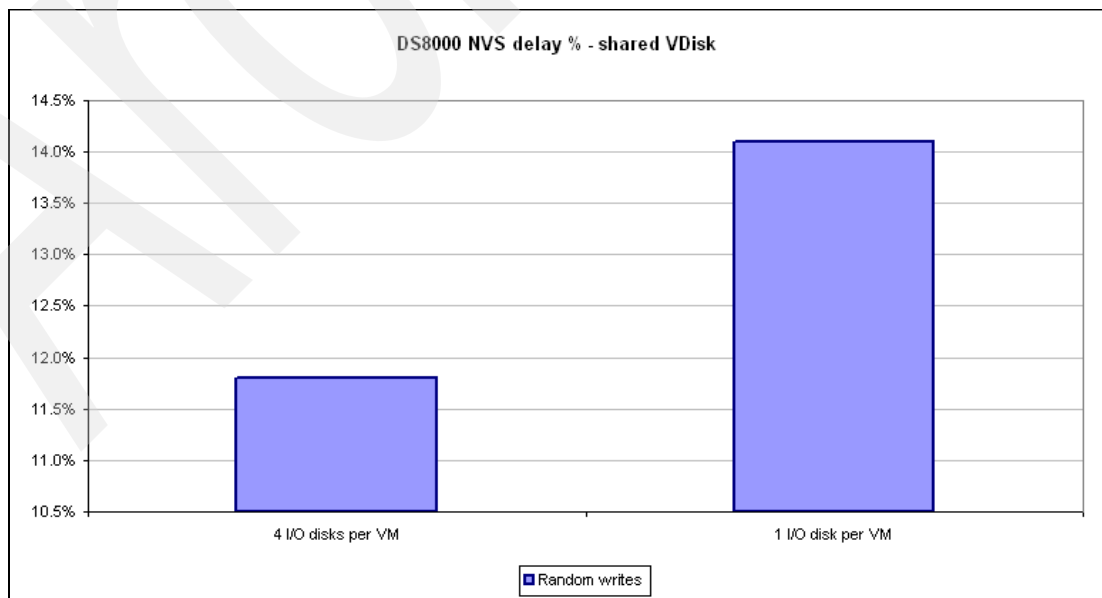


Figure 23 DS8000 NVS delays with four versus one I/O disks for random writes on a shared VDisk

### SVC cache hit rates

Figure 24 only applies to the R/W and read-only test cases.

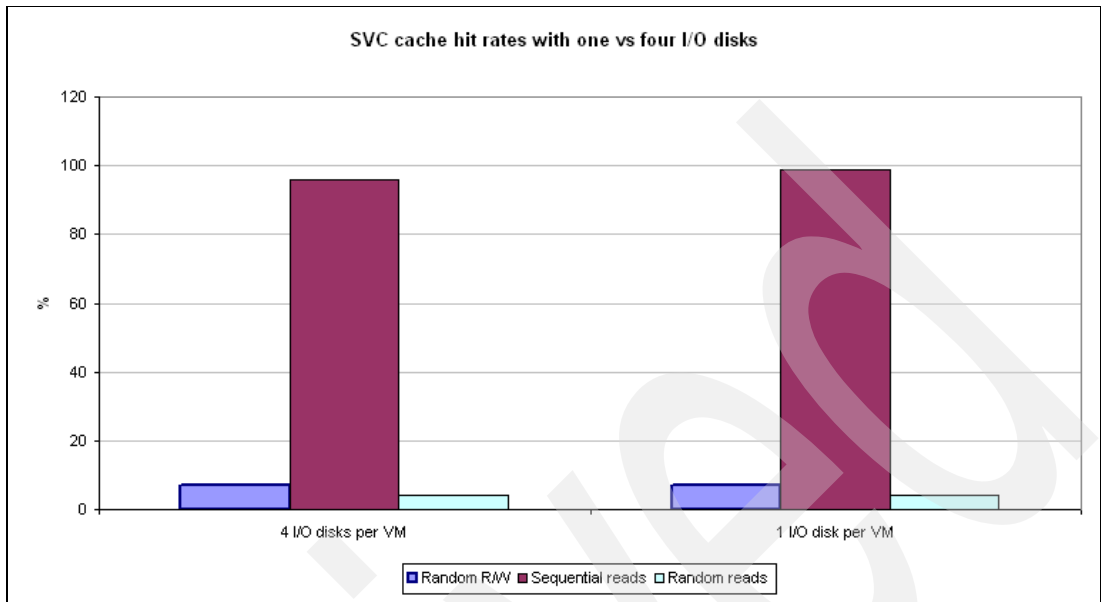


Figure 24 SVC cache hit rates with four versus one disks

It shows the benefit of the prefetch cache algorithms in the SVC during sequential reads.

Figure 25 shows the equivalent tests with the 800 GB shared VDisk.

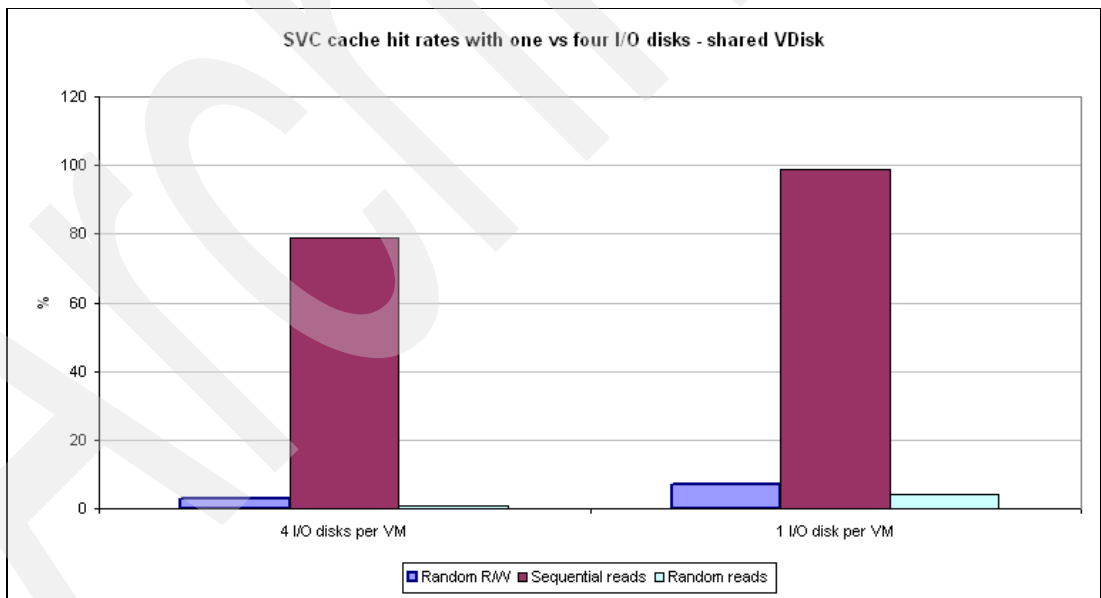


Figure 25 SVC cache hit rates with four versus one disks on a shared VDisk

### DS8000 cache hit rates

Figure 26 only applies to the R/W and read-only test cases.

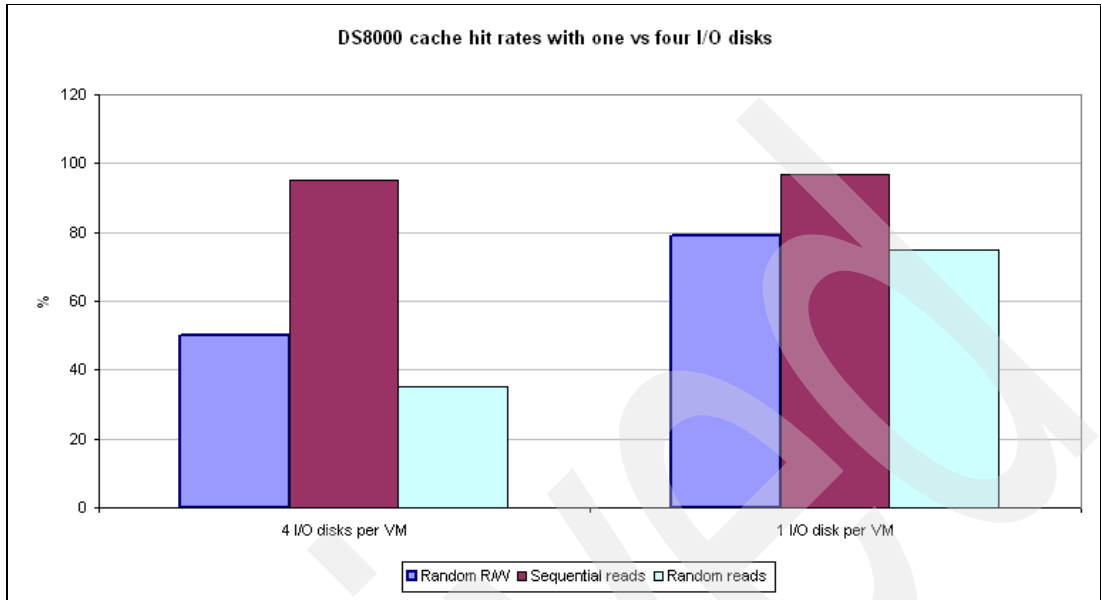


Figure 26 DS8000 cache hit rates with four versus one disks

It shows the benefit of the prefetch cache algorithms in the DS8000 during sequential reads.

Figure 27 shows the equivalent tests with the 800 GB shared VDisk.

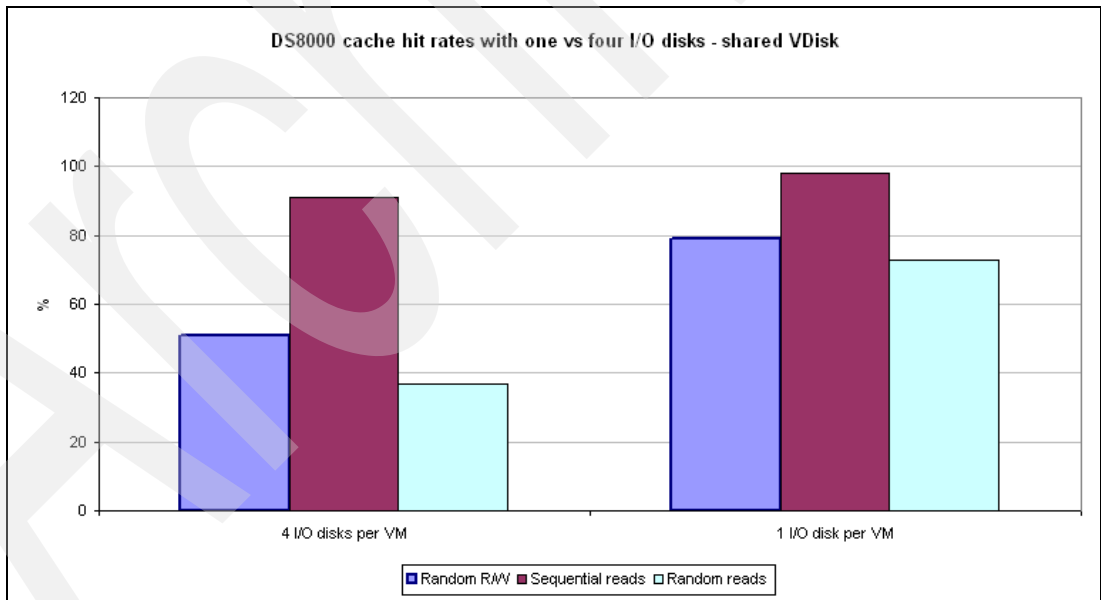


Figure 27 DS8000 cache hit rates with four versus one disks on a shared VDisk

### ESX host CPU loading

Figure 28 shows the CPU utilization of the four Xeon processors on one of the xSeries servers when running the test cases. It shows the average utilization to be just over 90%, and peaking at 100%, so the servers are effectively running at their maximum CPU load.

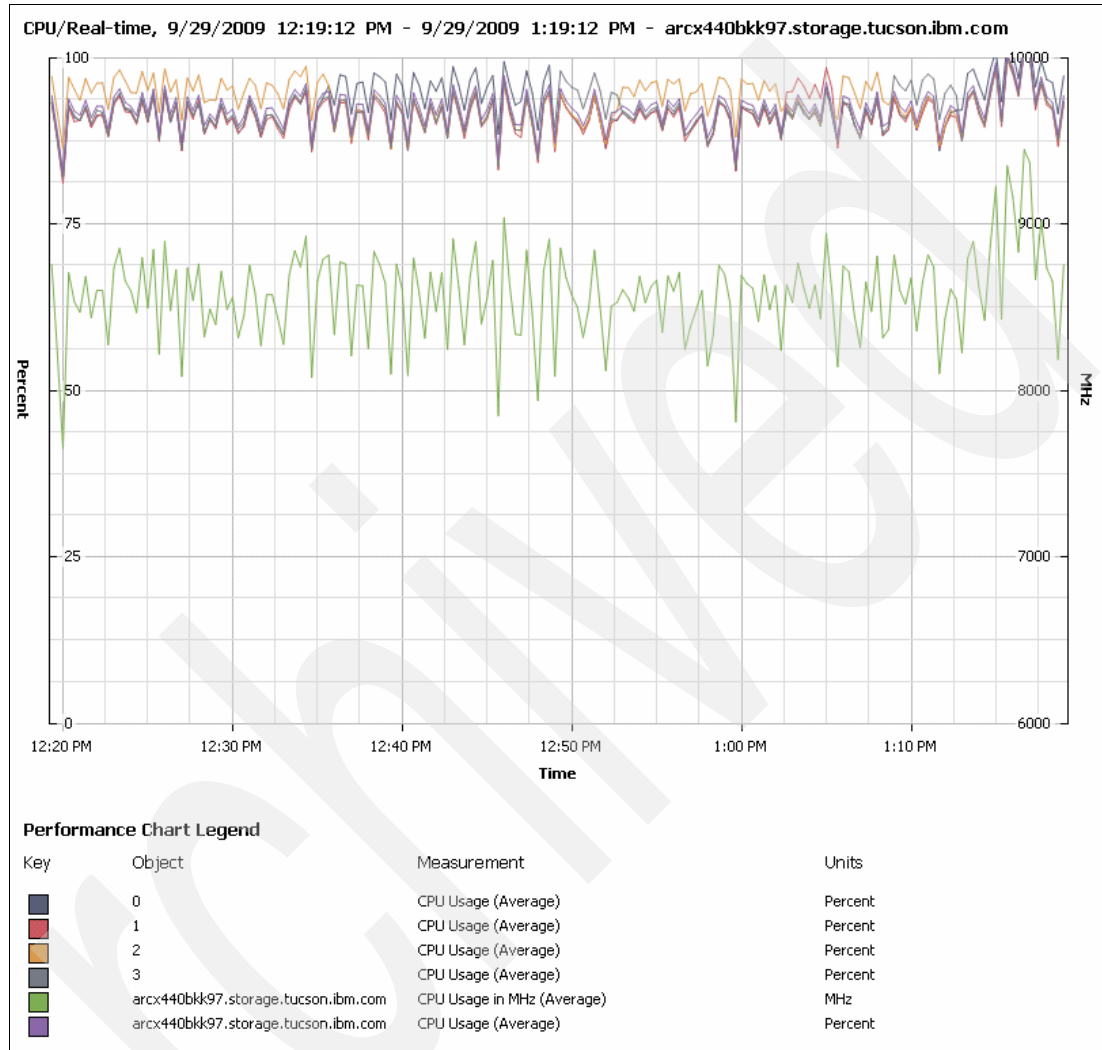


Figure 28 Single ESX host CPU load during performance testing

### Conclusions

We reach the following conclusions:

- ▶ Performance decreased when moving from four I/O disks to one I/O disk. This was probably due to increased queuing for the single I/O disk within VMware.
- ▶ There was very little difference between the tests with multiple VDIs and the tests with one large VDI, which was not the expected result. Looking at the ESX CPU loading in Figure 28, it becomes clear that the tests were CPU constrained and hence unable to generate sufficient I/O to exceed the performance of a single VDI.

### Test group 3

This set of tests is intended to determine how many VMware metadata operations can be performed in a given time without causing significant impact to data I/O. The configuration used is described in “Test group 3” on page 13 and shown in Figure 29.

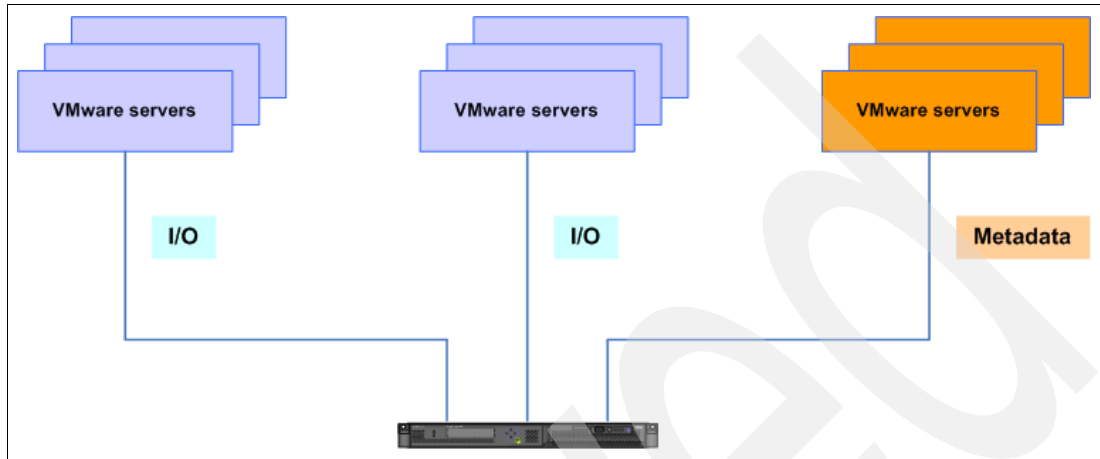


Figure 29 I/O and metadata from separate ESX hosts

#### ***Five second step-down with a separate metadata host***

The following tests started metadata operations to establish a baseline. A virtual disk create/delete operation was performed repeatedly at a fixed interval of 60 seconds for 10 minutes. After this task completed, the interval was reduced by 5 seconds for the next 10 minutes. This delay step-down continued until the delay was zero. After this, the metadata operations were stopped.

Figure 30 shows that the I/O performance only starts to be impacted once the delay interval reaches about 5 seconds.

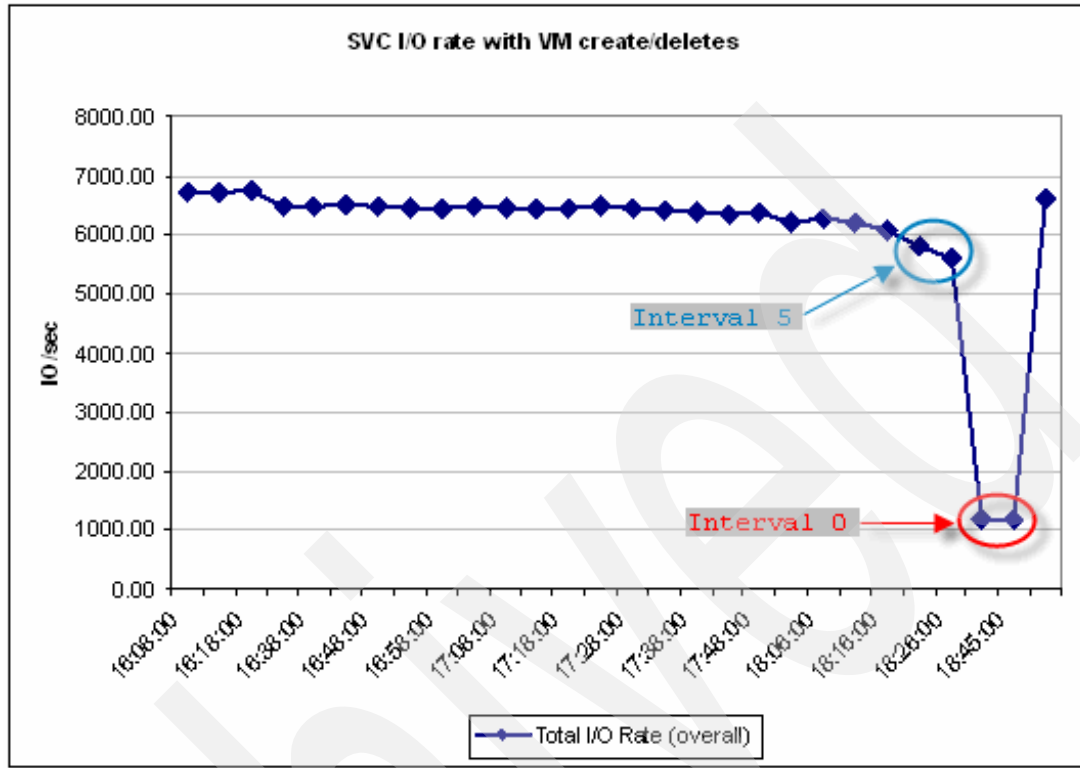


Figure 30 SVC I/O rate with virtual disk create/delete operations: 5 second step-down

The graph of the SVC data rate shows the same result, as shown in Figure 31.

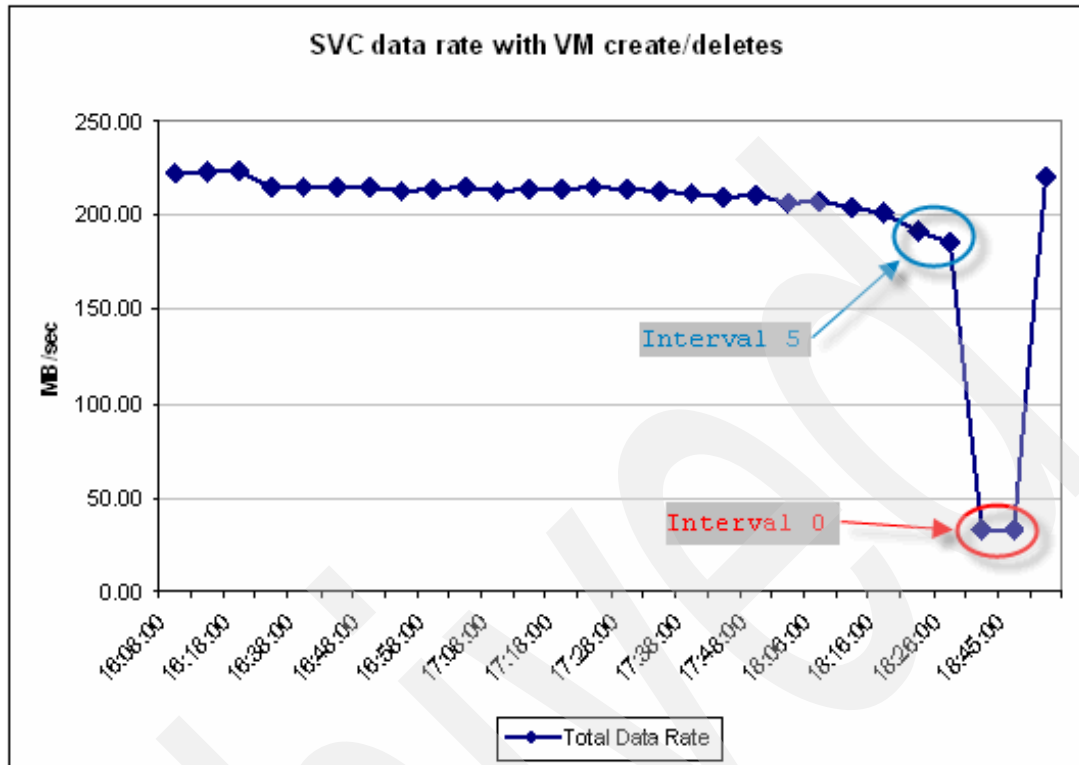


Figure 31 SVC data rate with virtual disk create/delete operations: 5 second step-down

Analysis of the SCSI commands issued by ESX to the SVC showed that every virtual disk create/delete cycle resulted in 23 pairs of SCSI-2 reserve/release commands being processed, along with up to 10 reservation conflicts for each of the other two ESX hosts for each cycle. While a single reserve/release takes only a few milliseconds, a stream of 46 such commands add up to a significant amount of time, hence the major performance impact once the delay interval reaches zero.

### ***One second step-down with a separate metadata host***

It is obvious that a reduction interval of five seconds is too large, so the test was rerun starting with a 20 second interval, reducing by one second every 10 minutes.



Figure 32 shows that the I/O performance starts to be impacted once the delay interval reaches six seconds, but only becomes significant at two seconds or less.

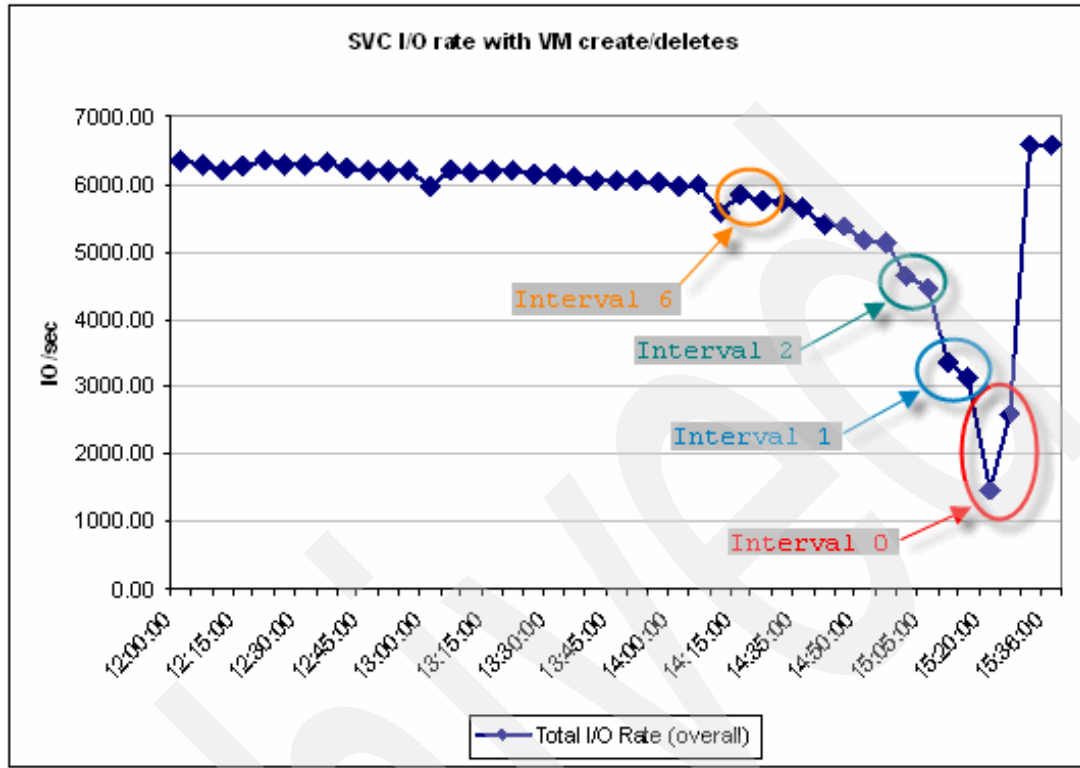


Figure 32 SVC I/O rate with virtual disk create/delete operations: 1 second step-down

The graph of the SVC data rate shows the same result, as shown in Figure 33.

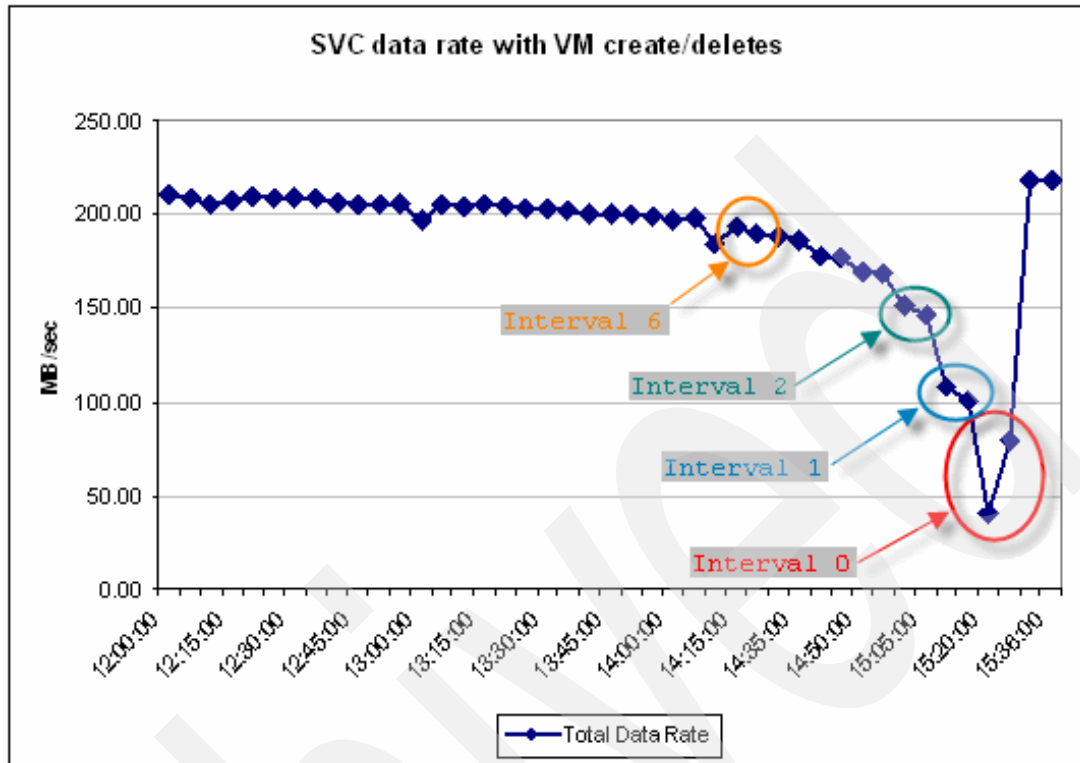


Figure 33 SVC data rate with virtual disk create/delete operations: 1 second step-down

Because the only difference here is the rate of reduction in the delay interval, the same number of SCSI-2 reserve/release command pairs and reservation conflicts are seen for each cycle. This explains why the performance impact becomes so great as the delay interval approaches zero, because the time for 46 reserve/release commands becomes a significant percentage of the delay interval.

**Performing I/O and create/delete on the same ESX host**

So far, all metadata operations have been performed from the third ESX host, which means that the ESX hosts running I/O will receive reservation conflicts while the SCSI-2 reserve is held. The following charts show the results of repeating the previous test with a one second step-down, but issuing the metadata requests on one of the two ESX hosts running I/O (the third ESX hosts is unused, as shown in Figure 34). The aim is to determine if this lessens the impact on I/O for the VMs on that host.

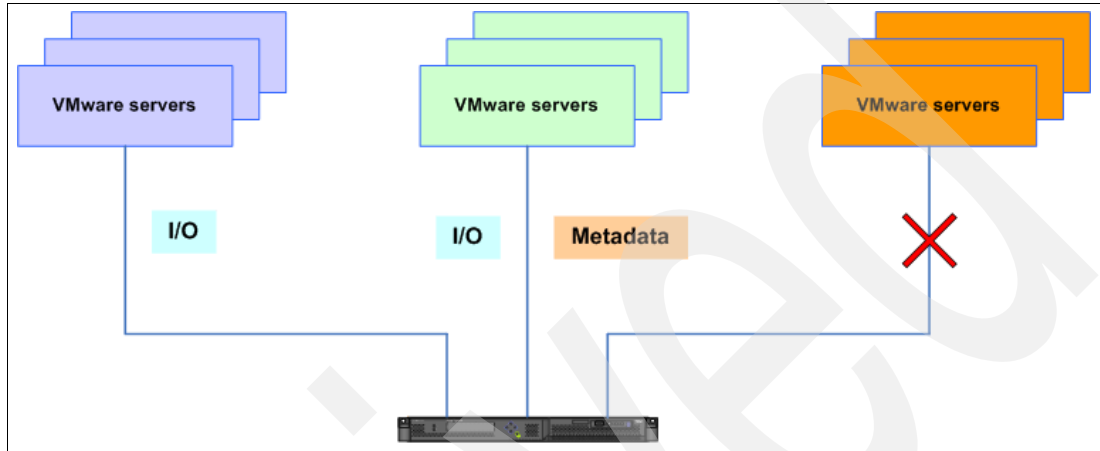


Figure 34 I/O and metadata from the same ESX host

Figure 35 shows that the I/O performance again starts to be impacted once the delay interval reaches six seconds, but only becomes significant at two seconds or less.

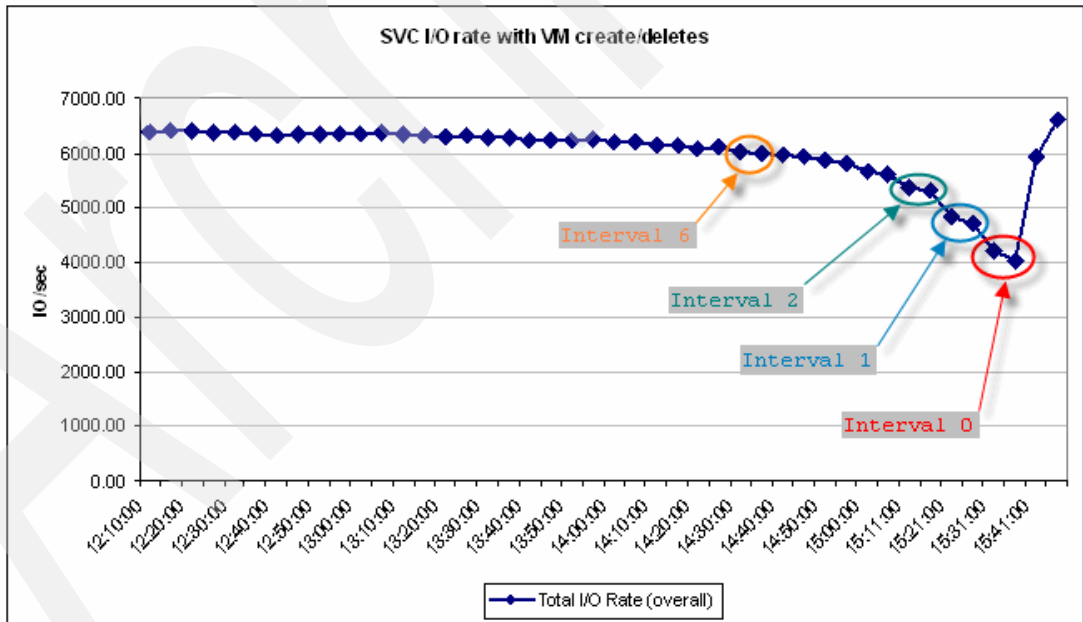


Figure 35 SVC I/O rate with virtual disk create/delete operations using two hosts

Again, the graph of the SVC data rate shows the same result, as shown in Figure 36.

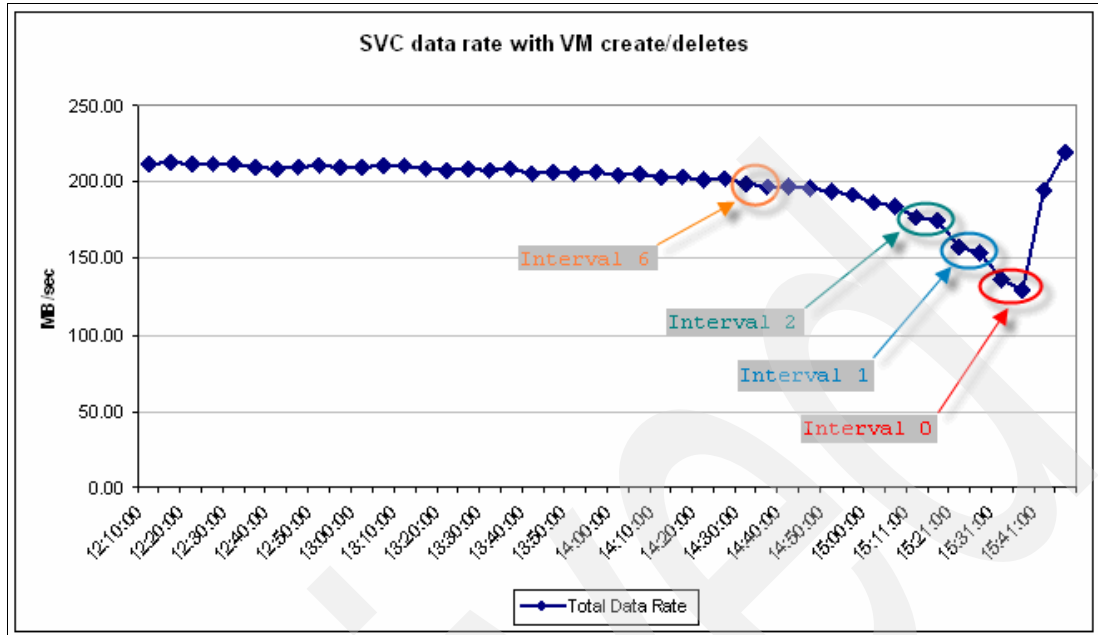


Figure 36 SVC data rate with virtual disk create/delete operations using two hosts

### Comparing shared and non-shared metadata hosts

Figure 37 compares the results for two ESX hosts (shared metadata and I/O data) and three ESX hosts (separate metadata).

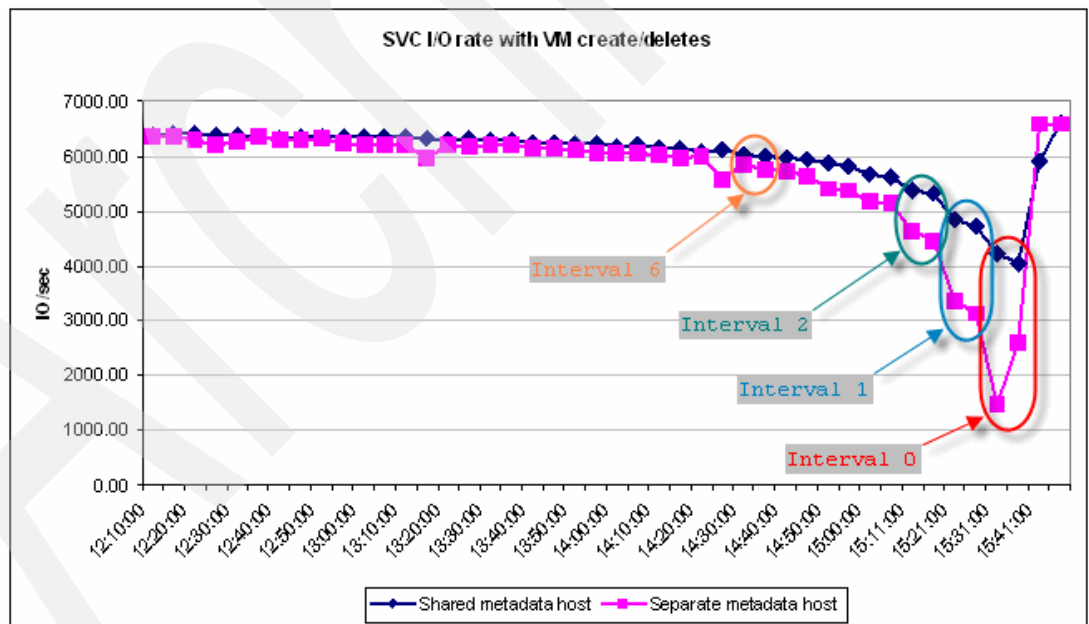


Figure 37 SVC I/O rate with virtual disk create/delete operations with shared versus non-shared

Although the shape of the two curves is similar, the rate of decline is much less in the shared case. This shows that there is less impact when the virtual disk create/delete operations are run from one of the hosts performing I/O, because that host is no longer experiencing reservation conflicts.

**Note:** If there were many ESX hosts sharing the VDisk, then the benefit of issuing the metadata operations from one of them would be less than in this example with only two hosts.

### **Conclusions**

We reach the following conclusions:

- ▶ I/O performance will degrade if metadata operations are performed frequently.
- ▶ Metadata operations do not impact I/O for the ESX host they are issued from because that host will not experience reservation conflicts.
- ▶ A single metadata operation can result in multiple reserve/release commands being issued.

## **Summary**

The various tests described in this IBM Redpapers publication looked at two different aspects of using VMware with the SVC, namely how different ways of sharing VDIs can affect VM performance, and how VMware metadata operations can impact performance.

### **VDisk options**

The expectation was that the test results presented here would demonstrate the benefits of spreading host I/O across multiple SVC nodes using carefully assigned paths. However, this has not been clearly shown due to the limited amount of server capacity available for the tests. If more servers, and hence more CPU capacity, had been available, the tests should have shown the expected results.

Two conclusions can be drawn from these results:

1. In many cases, just allocating a few large LUNs and sharing them for multiple VMs will work, as the SVC has the performance to cope with large loads.
2. In cases with a higher than average I/O workload, extra disk performance can be obtained by ensuring optimal pathing and distribution of VDisk workload across multiple SVC nodes.

The VMs in this test were unusual in that they were purely generating I/O as fast as possible. Because most VM workloads are not in this situation, many more ESX hosts and VMs can be supported by the SVC than are shown here.

### **Metadata impact**

The results of the SCSI-2 reserve/release tests show that performance only becomes an issue when repeated metadata events occur at short intervals on the order of two seconds. However, since different metadata events can cause a different number of SCSI-2 reserve/release commands to be generated, the interval before performance is impacted could vary.

It is important to understand that such metadata operations have the potential to impact I/O performance, and to schedule such operations accordingly.

## Recommendations

- ▶ Always follow the previously mentioned best practice for configuring MDisks on the back-end RAID storage.
- ▶ With ESX V3.5, use the Fixed multipathing policy and ensure that the paths are evenly distributed across all HBAs and SVC ports. Also, be sure to select the SVC preferred node for each VDisk.
- ▶ With vSphere V4 or ESXi V4, use the Round Robin multipathing policy, which should distribute paths automatically.
- ▶ Unless you know your applications will generate a great deal of I/O, start with a simple configuration of VMFS on a single 500 GB VDisk, measure your performance, and adjust as necessary.
- ▶ For configurations that perform a great deal of metadata operations (such as Snapshot™), place fewer VMs on each VMFS, using more and smaller VDIsks if necessary.
- ▶ Size your LUNs by expected peak I/O rates rather than simply capacity.

## Related material

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM SVC and Brocade Disaster Recovery Solutions for VMware*, REDP-4626
- ▶ *VMware Multi-pathing with the SVC, and the Causes of SCSI-2 Reservation Conflicts*, REDP-4627

## The team who wrote this IBM Redpapers publication

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

**Steve Garraway** is an IT specialist with the Storage and SAN Services team at the IBM Hursley Laboratory in the United Kingdom. He has 22 years of experience in storage management, mainly with z/VM® and IBM Tivoli Storage Manager, and has worked for IBM for 23 years. His areas of expertise include IBM System z® disk and tape storage, McDATA and Brocade SANs, and the SAN Volume Controller. He has previously co-authored four IBM Redbooks publications on enterprise disk and SAN. Steve is a Brocade Certified Fabric Professional.

**Jon Tate** is a Project Manager for IBM System Storage SAN Solutions at the International Technical Support Organization, San Jose Center. Before joining the ITSO in 1999, he worked in the IBM Technical Support Center, providing Level 2 and 3 support for IBM storage products. Jon has 24 years of experience in storage software and management, services, and support, and is both an IBM Certified IT Specialist and an IBM SAN Certified Specialist. He is also the UK Chairman of the Storage Networking Industry Association.

Thanks to the following people for their contributions to this project:

Doug Johnston  
Hemanth Kalluri  
Troy Lee  
IBM Open Systems Lab, San Jose

Bill Passingham  
Steve White  
Barry Whyte  
IBM Hursley Laboratory, UK

Archived

Archived



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4601-00 was created or updated on December 18, 2009.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an email to:  
[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.



## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DS8000®	Redpapers™	Tivoli®
eServer™	Redbooks (logo) ®	xSeries®
IBM®	System Storage™	z/VM®
Redbooks®	System x®	
Redpaper™	System z®	

The following terms are trademarks of other companies:

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

VMotion, VMware, the VMware "boxes" logo and design are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel Xeon, Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.