Scalable Storage Performance

VMware® ESX 3.5

VMware ESX enables multiple hosts to share the same physical storage reliably through its highly optimized storage stack and VMware Virtual Machine File System (VMFS). Centralized storage of virtual machines using VMFS provides more control and flexibility. It also enables such unique virtualization capabilities as live migration (VMware VMotion), VMware Distributed Resource Scheduler, VMware High Availability, and clustering. To gain the greatest advantage from shared storage, you must understand the storage performance limits of a given physical environment to ensure that you do not overcommit resources.

To make the most effective use of your storage, you need to understand:

- How many virtual machines can share the same LUN
- How SCSI reservations affect shared storage performance in ESX
- How many LUNs and VMFS file systems can be configured on a single ESX host

This paper presents the results of our studies on storage scalability in a virtual environment with many ESX hosts, many LUNs, or many of both. It examines the effects of I/O queuing at various layers in a virtual infrastructure as more and more virtual machines share the same storage. It considers the effects of SCSI reservations on virtual machine I/O performance. And it looks at ways to mitigate bandwidth bottlenecks when multiple LUNs are connected to a single ESX host. It provides recommendations you can follow to avoid overcommitting storage resources.

This study covers the following topics:

Measures of Scalability

Two distinct measurements are used when evaluating scalability—throughput and latency.

Throughput

Throughput is the amount of data transferred in a unit of time and is most commonly measured in kilobytes per second (KBps) or megabytes per second (MBps). The throughput in an environment depends on many factors, related to both hardware and software. Among the important factors are Fibre Channel link speed, number of outstanding I/O requests, number of disk spindles, RAID type, SCSI reservations, and caching or prefetching algorithms.

Latency

Latency is the time taken to complete an I/O request and is most commonly measured in milliseconds (msec). Because multiple layers are involved in a storage system, each layer through which the I/O request passes might add its own delay. Latency depends on many factors, including queue depth or capacity at various levels; I/O request size; disk properties such as rotational, seek, and access delays; SCSI reservations; and caching or prefetching algorithms.

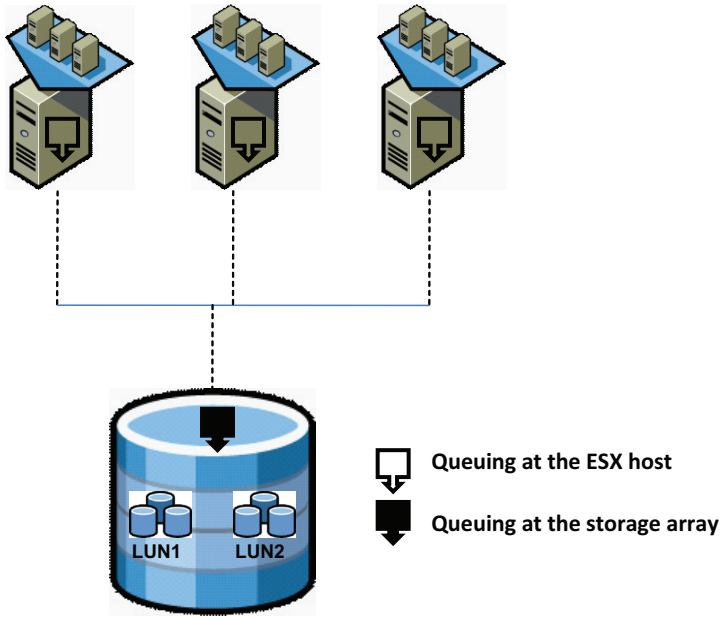
Factors Affecting Scalability of ESX Storage

Our tests explored three key factors that affect the scalability of storage in an ESX environment—the number of active commands, SCSI reservations, and total available link bandwidth.

Number of Active Commands

The SCSI protocol allows multiple commands to be active on a LUN at the same time. SCSI device drivers have a configurable parameter called the LUN queue depth that determines how many commands can be active at one time to a given LUN. QLogic Fibre Channel HBAs support up to 255 outstanding commands per LUN, and Emulex HBAs support up to 128. However, the default value for both drivers is set to 32. If an ESX host generates more commands to a LUN than the LUN queue depth, the excess commands are queued in the ESX kernel, and this increases the latency.

Figure 1. Command Queues on Storage Arrays and ESX Hosts



When you consider a cluster of ESX hosts connected to a storage array, SCSI command queuing can occur mainly in two places—on the storage array and on the ESX host, as shown in Figure 1.

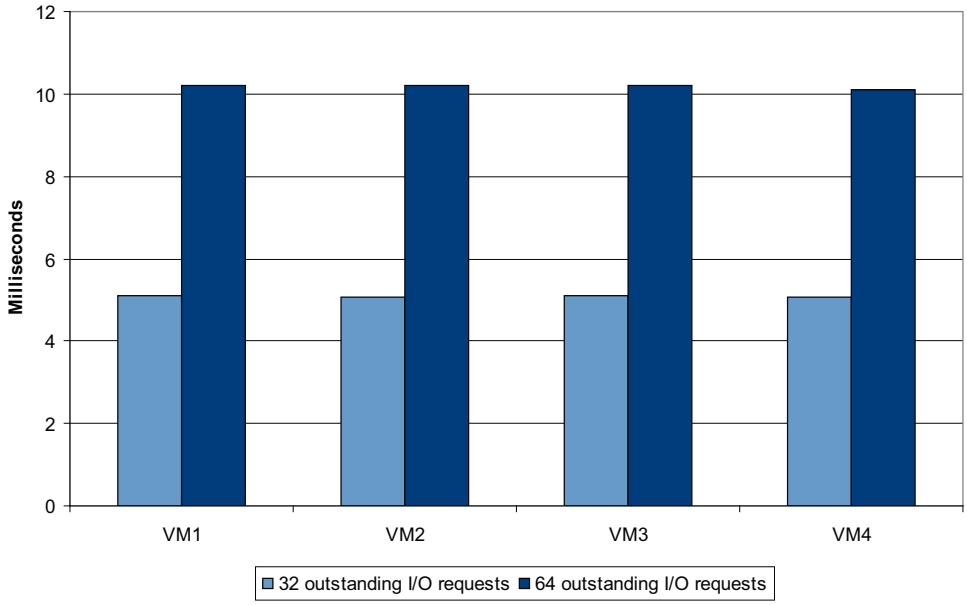
Command Queuing on the ESX Host

SCSI device drivers have a configurable parameter called the LUN queue depth that determines how many commands to a given LUN can be active at one time. The default in ESX is 32. If an ESX host generates more commands to a LUN than the LUN queue depth, the excess commands are queued in the ESX kernel, and this increases the latency. The queue depth is per-LUN, and not per-initiator. The initiator (or the host bus adapter) supports many more commands (typically 2,000 to 4,000 commands per port). This means that if two virtual machines have their virtual disks on two different LUNs, each virtual machine can generate as many active commands as the LUN queue depth. But if the two virtual machines have their virtual disks on the same LUN (VMFS volume), the total number of active commands that the two virtual machines combined can generate without incurring queuing delays is equal to the LUN queue depth. More specifically, when virtual machines share a LUN, the total number of outstanding commands permitted from all virtual machines to that LUN is

governed by the `Disk.SchedNumReqOutstanding` configuration parameter that can be set using VirtualCenter. If the total number of outstanding commands from all virtual machines exceeds this parameter, the excess commands are queued in the ESX kernel.

Figure 2 shows the results of a test in which four virtual machines generate I/O to the same LUN. In the first case, each virtual machine generates eight outstanding commands. In the second case, each virtual machine generates 16 commands. In the second case, because the total number of outstanding commands to the LUN exceeds 32 (which is the default LUN queue depth), the latency is double that of the first case. The latency doubles because the excess commands must be queued in the ESX kernel until the currently issued commands are completed.

Figure 2. Latency for Multiple Virtual Machines Using the Same LUN



Recommendation for Reducing Queuing on the ESX Host

To reduce latency, ensure that the sum of active commands from all virtual machines does not consistently exceed the LUN queue depth. Either increase the queue depth as shown in the VMware Infrastructure 3 *Fibre Channel SAN Configuration Guide* (the maximum recommended queue depth is 64) or move the virtual disks of some virtual machines to a different VMFS volume. You can find the guide at http://www.vmware.com/pdf/vi3_35/esx_3/r35/vi3_35_25_san_cfg.pdf.

Also make sure to set the `Disk.SchedNumReqOutstanding` parameter to the same value as the queue depth. If this parameter is given a higher value than the queue depth, it is still capped at the queue depth. However, if this parameter is given a lower value than the queue depth, only that many outstanding commands are issued from the ESX kernel to the LUN from all virtual machines. The `Disk.SchedNumReqOutstanding` setting has no effect when there is only one virtual machine issuing I/O to the LUN.

Command Queuing on the Storage Array

To understand the impact of queuing on the storage array, consider a case in which the virtual machines on an ESX host can generate a constant number of SCSI commands equal to the LUN queue depth. If multiple such ESX hosts share the same LUN, SCSI commands to that LUN from all hosts are processed by the same storage processor on the storage array (in the case of active/passive arrays) or by a group of storage processors (in the case of active/active arrays). Strictly speaking, an array storage processor running in target mode might not have a per-LUN queue depth, so it might issue the commands directly to the disks. But if the number of active commands to the shared LUN is too high, multiple commands begin queuing up at the disks, resulting in high latencies.

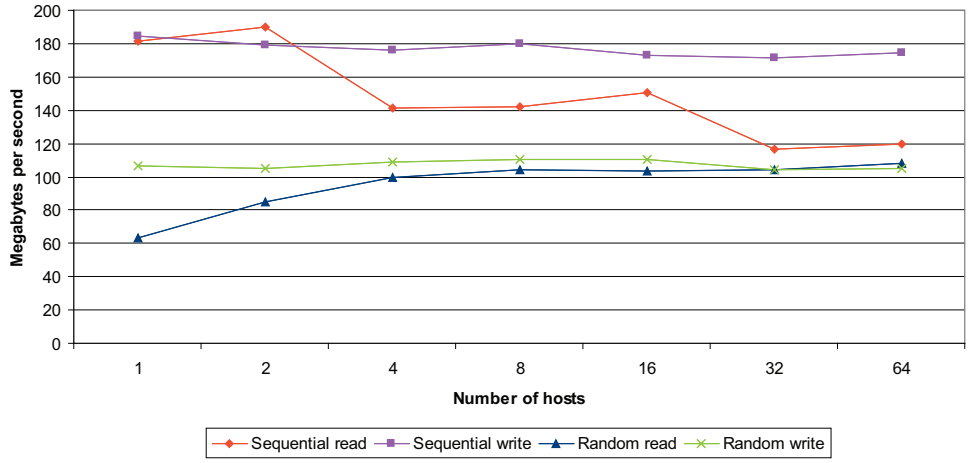
We tested the effects of running an I/O intensive workload on 64 ESX hosts sharing a single nonspanned VMFS volume (that is, a VMFS volume on a single LUN).

We carried out the test using the configuration shown in Table 1.

Table 1. Configuration for I/O Workload Test

Number of ESX hosts	64
Number of VMFS volumes	1
LUNs per VMFS volume	1
LUN properties	RAID0, 15 disks, 128GB
Link speeds	2Gbps from the ESX host to the Fibre Channel switch 4Gbps from the Fibre Channel switch to the storage array
Workload	IOmeter, 32KB block size
Virtual machines per host	1
Commands per virtual machine	32

Figure 3. Aggregate Throughput for I/O Workload Test



As Figure 3 shows, except for sequential reads, there is no drop in aggregate throughput as we scale the number of hosts. The reason sequential read throughput drops is that the sequential streams coming in from different ESX hosts are intermixed at the storage array, thus losing their sequential nature. Writes generally show better performance than reads because they are absorbed by the write cache and flushed to disks in the background.

Figure 4. Average Latency for I/O Workload Test

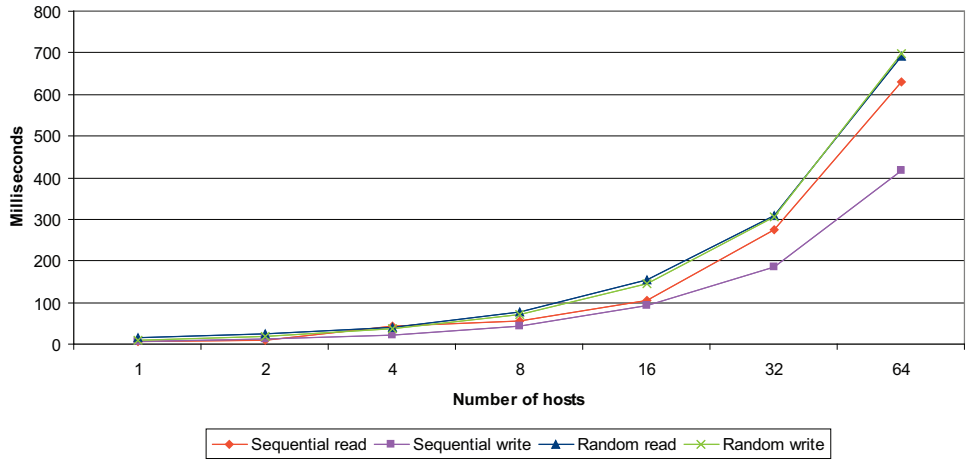


Figure 4 shows the results of commands from all ESX hosts issued to the shared LUN on the storage array. Each ESX host generates 32 commands, thus with eight hosts the LUN has 256 outstanding commands. With more than eight hosts, latencies increase to more than 50 milliseconds and could affect applications that are sensitive to latencies, although there is no drop in aggregate throughput.

NOTE The latencies shown in Figure 4 are specific to our experimental configuration and depend on a variety of factors including the number of disks backing the LUN, the speed of the disks (rotational speed and interconnect speed), and the storage technology (magnetic disks or solid state disks).

Our test uses a specific configuration with an aggressive I/O rate. Virtual machines deployed in typical customer environments may not have as high an I/O rate and therefore may be able to scale further. In general, because of varying block sizes, access patterns, and number of outstanding commands, the results you see in your VMware environment depend on the types of applications running. The results also depend on the capabilities of your storage and whether it is tuned for the block sizes in your application.

Recommendation for Reducing Queuing on the Storage Array

The number of virtual machines (and hence the number of ESX hosts) that can share a single VMFS volume depends on the I/O activity of the virtual machines and also on the capabilities of the storage array. In general, there are two constraints that govern how virtual machines are distributed among ESX hosts and how many LUNs (or VMFS volumes) are needed. The constraints are:

- A maximum LUN queue depth of 64 per ESX host. If an ESX host has exclusive access to a LUN, you can increase the queue depth to 128 if an application really demands it. However, if multiple ESX hosts share the same LUN, it is not recommended to have a queue depth setting of more than 64, because it is possible to oversubscribe the storage array if it is short of hardware capabilities.
- A maximum number of outstanding I/O commands to the shared LUN (or VMFS volume) that depends on the storage array. This number must be determined for a particular storage array configuration supporting multiple ESX hosts. If the storage array has a per-LUN queue depth, exceeding this value causes high latencies. If the storage array does not have a per-LUN queue depth, the bottleneck is shifted to the disks, and latencies increase. In either case, it is important to ensure that there are enough disks to support the influx of commands. It is hard to recommend an upper threshold for latency because it depends on individual applications. However, a 50 millisecond latency is high enough for most applications, and you should add more physical resources if you reach that point.

Table 2 illustrates how to compute the maximum number of virtual machines per LUN. The numbers in the *n* column represent the maximum outstanding number of I/O commands recommended for a LUN on a particular storage array. This is an array-specific parameter that has to be determined in consultation with the storage vendor. The numbers in the *a* column represent the average number of active I/O commands per virtual machine sharing the same LUN.

Table 2. Calculating Load on a VMFS volume for Sample Configurations

Maximum Outstanding I/O Recommended for Array, per LUN (*n*)	Average Active SCSI Commands per Virtual Machine to the Shared VMFS (*a*)	LUN Queue depth on each ESX Host (*d*)	Maximum Number of Virtual Machines per ESX Host on the Shared VMFS *m=(d/a)*	Maximum Number of Virtual Machines on the Shared VMFS (*n/a*)
256	4	32	8	64
256	4	64	16	64
1024	4	64	16	256
256	32	32	1	8
1024	32	32	1	32
1024	1	32	32	1024

You can determine the average SCSI commands per virtual machine to the shared VMFS as follows: Run the ESX command line utility `esxtop` in batch mode for at least an hour during the steady-state operation of the

virtual machine with QSTATS enabled in disk view. After collecting the batch output, look for the following keywords using a CSV-format file editor or program (for example, Perfmon in Windows):

```
\\<host>\Physical Disk(vmhba<A>:<C>:<T>:<L>:<W>)\Active Commands
\\<host>\Physical Disk(vmhba<A>:<C>:<T>:<L>:<W>)\Queued Commands
```

Or:

```
\\<host>\Physical Disk(WD-vmhba<A>:<T>:<L>-<W>)\Active Commands
\\<host>\Physical Disk(WD-vmhba<A>:<T>:<L>-<W>)\Queued Commands
```

In the lines above, <A>=Adapter, <C>=Channel, <T>=Target, <L>=LUN, and <W>=World ID of the virtual machine. The sum of the active and queued commands for that virtual machine give the total number of outstanding commands issued by that virtual machine.

The calculations in Table 2 assume that there is only one path to the LUN or the array is active/passive in nature. The value of *m* will be a multiple of the number of adapters on the ESX host if the same LUN has multiple paths going through those different adapters and a multipath plug-in is used in ESX (the storage array is active/active).

The maximum number of virtual machines per ESX host on the shared VMFS includes only those virtual machines residing on the shared VMFS volume. You can run additional virtual machines residing on other VMFS volumes on the same host, for a maximum of 128 virtual CPUs per host on VMware ESX 3.5.

SCSI Reservations

VMFS is a clustered file system and uses SCSI reservations as part of its distributed locking algorithms. Administrative operations, such as creating or deleting a virtual disk, extending a VMFS volume, or creating or deleting snapshots, result in metadata updates to the file system using locks, and thus result in SCSI reservations. Reservations are also generated when you expand a virtual disk for a virtual machine with a snapshot. A reservation causes the LUN to be available exclusively to a single ESX host for a brief period of time. Although it is acceptable practice to perform a limited number of administrative tasks during peak hours, it is preferable to postpone major maintenance or configuration tasks to off-peak hours in order to minimize the impact on virtual machine performance.

You can take advantage of all the benefits of a clustered file system while still maintaining good overall performance if you have some insights into how VMFS locking works.

We conducted a test that shows the performance effects of SCSI reservations. Host1 is an ESX 3.5 host on which one virtual machine is generating SCSI commands to a VMFS volume for two minutes and recording throughput and latency. Host2 is another ESX 3.5 host sharing the VMFS volume with Host1. No virtual machines on Host2 are sending SCSI commands to the VMFS volume. Table 3 summarizes the results.

Table 3. Performance Effects of SCSI Reservations

	Throughput	Average Latency (msec)	Virtual Machine Create (sec)	Virtual Machine Delete (sec)
No virtual machine creation or deletion	45.2	22.1	NA	NA
Virtual machine creation or deletion from Host2	42.1	23.7	0.5	0.5
Virtual machine creation or deletion from Host1	44.5	22.5	15.7	16.0

The results in Table 3 show performance for three cases from the perspective of the virtual machine running on Host1.

- No SCSI reservations

When neither ESX host is creating or deleting virtual machines— that is, with only the single virtual machine running and generating SCSI commands to the shared LUN—the test produces baseline data for the configuration running uninterrupted.

- SCSI reservations generated by another host (Host2)

  When Host2 is creating and deleting virtual machines, it runs five iterations of creating and deleting a virtual machine on the shared VMFS volume. These operations involve metadata updates to the VMFS volume and thus Host2 issues SCSI reservations on the shared VMFS volume. These SCSI reservations cause a 7 percent drop in throughput as seen by the virtual machine on Host1 for the two-minute interval when it is generating SCSI commands. There is a corresponding increase of 7 percent in the average latency, because the SCSI commands from the virtual machine must be retried every time they encounter a reservation conflict on the shared LUN.

- SCSI reservations generated by the same host (Host1)

  When Host1 is creating and deleting virtual machines, two workloads are running on Host1—the virtual machine generating SCSI commands and the virtual machine creation and deletion operations. In this case, the SCSI commands from the virtual machine are not affected very much—just under 2 percent. On the other hand, the virtual machine creation and deletion takes much longer to complete. The reason for this difference from the previous case is that, because the SCSI reservations are being generated by Host1 itself, the SCSI commands from the virtual machine do not run into reservation conflicts, and thus the storage system can process the commands immediately. However, the commands from the virtual machine creation and deletion operation are going to the same LUN, thus it takes more time to finish creating and deleting the virtual disks on the shared storage.

Recommendations for Avoiding SCSI Reservation Conflicts

Two guidelines can help you avoid SCSI reservation conflicts.

- Perform administrative tasks such as creating and deleting virtual machines, deploying from templates, extending a VMFS volume, or creating and deleting snapshots at off-peak hours, if possible.
- If you must perform administrative tasks urgently, the impact on virtual machines depends on the specific ESX cluster where you must perform the tasks. If the ESX host on which you perform the administrative task also hosts I/O-intensive virtual machines, you see little to no impact on those virtual machines. They can continue to access the shared LUN, because SCSI reservations are issued by the same host. I/O-intensive virtual machines running on other ESX hosts are affected for the duration of the administrative task, because they are likely to encounter SCSI reservation conflicts. As a result, if you can choose the specific ESX host from which to run the administrative tasks, choose an ESX host that is running I/O-intensive virtual machines.

Total Available Link Bandwidth

If multiple I/O-intensive virtual machines are running on the same ESX host, be sure that you have enough Fibre Channel links of the proper capacity (1/2/4 Gbps) to all the VMFS volumes used by that ESX host.

Figure 5. Aggregate Throughput of Multiple I/O-Intensive Virtual Machines

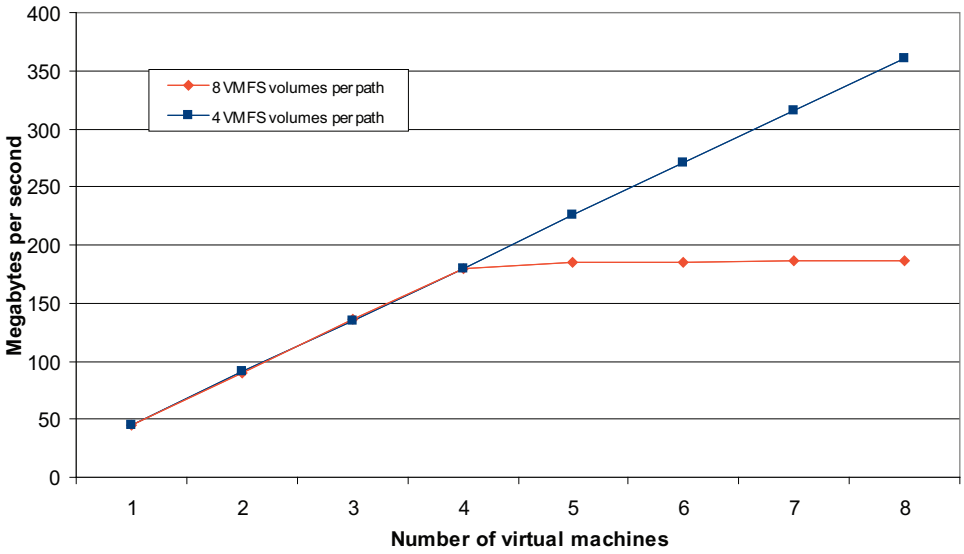


Figure 5 shows the results of running an increasing number of I/O-intensive virtual machines on a single ESX host. Each I/O-intensive virtual machine resides on a different VMFS volume and is driving about 45MBps of data to its respective VMFS volume. In one case, the active paths of all the VMFS volumes are routed via the same link (eight VMFS volumes per link). In the other case, an additional link is added from the host to the storage array and the active paths are configured with four active paths per link (four VMFS volumes per link).

As expected, with eight VMFS volumes per link, the throughput flattens out after four virtual machines because the 2Gbps link is saturated (an effective data rate of approximately 180MBps). With the second link and the paths rerouted, additional bandwidth is available and the aggregate throughput continues to scale.

Figure 6. Average Latency of Multiple I/O-Intensive Virtual Machines

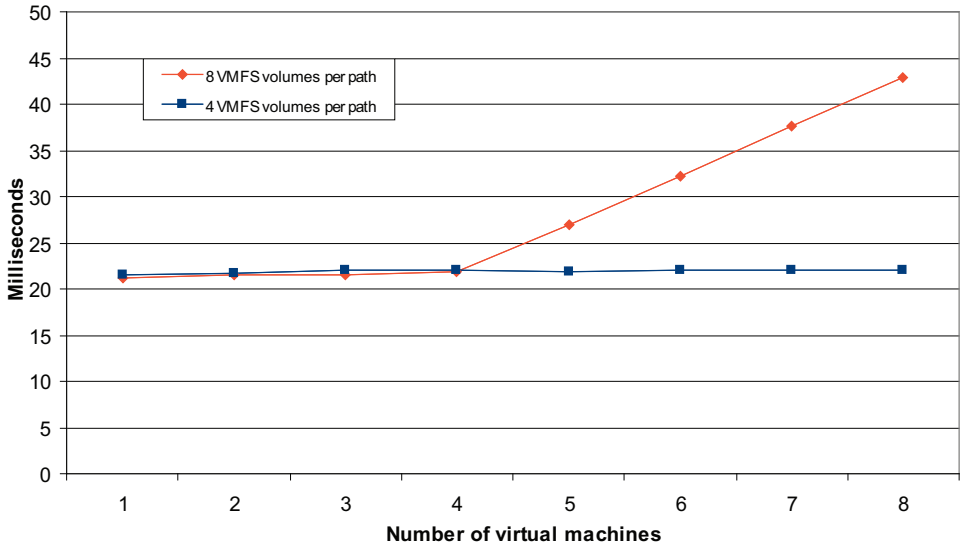


Figure 6 shows the effect on latency in the same test, running an increasing number of I/O-intensive virtual machines on a single ESX host. At eight VMFS volumes per link, and beyond four virtual machines, the commands from each virtual machine sit in the host bus adapter waiting for the link to be free. Note that this is not the queue depth issue that we covered earlier. As a result, the latencies increase with each additional virtual machine. On the other hand, with two links, the aggregate link bandwidth is adequate for the commands to be processed at wire speed.

Recommendation for Providing Adequate Link Bandwidth

You can connect as many as 256 LUNs per ESX host provided that you have enough Fibre Channel links to service the bandwidth needs of all virtual machines running on that particular ESX host, and also provided that you are within the CPU and memory constraints on that ESX host.

Spanned VMFS Volumes

A VMFS volume is said to be spanned if it includes multiple LUNs. Spanning is a good feature to use if you need to add more storage to a VMFS instance on the fly. VMFS spanning is a simple technology that does not involve striping or mirroring across the spanned volumes. Spanning adds capacity to an existing VMFS volume in the form of extents (most commonly a partition on a LUN), and VMFS manages the total available space across all extents as if they were a single device.

Predicting performance with spanned volumes is not straightforward because the user does not have control over how the data from various virtual machines is laid out on the different LUNs that form the spanned volume.

For example, consider a spanned VMFS volume with two LUNs. Both LUNs are 100GB in size. Also consider a case in which there are two virtual machines on this spanned VMFS, and the sum total of their sizes is 150GB. It is not possible for the user to determine the contents of each LUN in the spanned volume directly. Hence it is not straightforward to determine the performance properties of this configuration.

The following are some performance implications and best practices for a spanned volume:

- If multiple virtual machines reside on the same spanned VMFS volume, it is not clear how many of those virtual machines share the same LUNs, hence it is not straightforward to determine the optimal queue depth and `Disk.SchedNumReqOutstanding` parameters. If you see consistent queuing of commands on any LUN that is part of the spanned volume, the best option is to increase the adapter queue depth on the ESX host and adjust `Disk.SchedNumReqOutstanding` accordingly.
- When you add capacity to a spanned VMFS volume, the size of the new LUN might influence performance. If a very big LUN is added to the spanned volume, it could potentially hold many new virtual machines, in turn creating more contention on the newly added LUN than would have been the case with a smaller LUN.
- Spanned VMFS volumes might improve performance in certain cases. As mentioned earlier, VMFS metadata updates usually involve locking files and hence result in SCSI reservations. VMFS is designed in such a way that only the first LUN in a spanned VMFS volume needs to be locked by the SCSI reservation. Hence virtual machines that do not have their data on the first LUN are not affected by storage or VMFS administrative tasks.
- Mixing storage devices of different performance characteristics on the same spanned volume could cause an imbalance in virtual machine performance because a virtual machine's blocks could be allocated across device boundaries, and each device might have a different queue depth.

Summary of Storage Scalability Recommendations

This section summarizes the recommendations presented in this study.

How Many Virtual Machines Can Share the Same LUN

The number of virtual machines that can share the same LUN depends on how I/O-intensive the virtual machines are. The heavier the I/O activity per virtual machine, the smaller the number that can share a single LUN. Always follow two rules that should simplify the task of determining this number.

- LUN queue depth limit

  The sum of active SCSI commands from all virtual machines running on an ESX host sharing the same LUN should not consistently exceed the LUN queue depth configured on the ESX host, to a maximum of 64.

- Storage array limit

  Determine the maximum number of outstanding I/O commands to the shared LUN (or VMFS volume). This value is specific to the storage array you are using, and you might need to consult your storage array vendor to get a good estimate for the maximum outstanding commands per LUN. A latency of 50 milliseconds is usually a reliable indication that the storage array either does not have enough resources or is not configured optimally for its current use.

For more details, see "Number of Active Commands" on page 2.

How SCSI Reservations Affect Shared Storage Performance in ESX

The effect of SCSI reservations depends on how many virtual machines running on how many ESX hosts share the same LUN and what administrative tasks are being performed at the same time on the shared LUN. Keep in mind that you usually do not have to worry about SCSI reservations if

- No storage or VMFS administrative tasks are being performed, as mentioned in "SCSI Reservations" on page 6
- No VMFS volumes are shared by multiple hosts

The impact of SCSI reservations depends on the number and nature of storage or VMFS administrative tasks being performed. Keep the following points in mind:

- The longer an administrative task (for example, creating a virtual machine with a larger disk or cloning from a template that resides on a slow NFS share) runs, the longer the virtual machines are affected. Also, the time to reserve and release a LUN is highly hardware and vendor dependent.
- Running administrative tasks from a particular ESX host does not have much impact on the I/O-intensive virtual machines running on the same ESX host.

For more details, see "SCSI Reservations" on page 6.

How Many LUNs or VMFS File Systems Can Be Connected to a Single ESX Host

A maximum of 256 LUNs can be connected to an ESX host as long as no single link becomes the bottleneck and there are enough CPU and memory resources on that ESX host to run the corresponding virtual machines. The number of VMFS volumes depends on how many LUNs are configured, with the restriction that a single VMFS volume can span a maximum of 32 LUNs.

Conclusion

A virtualized environment makes effective use of available resources, but at the same time it can impose more load on the storage infrastructure because of increased consolidation levels. An I/O command generated in a virtualized environment must pass through extra layers of processing that enable all the useful features of virtualization. It is important to understand the potential bottlenecks at various layers and make the necessary configuration changes to get optimal storage performance.

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

Revision 20080603 Item: PS-055-INF-01-01