Technical Report

VMware vSphere 4.1 Storage Performance: Measuring FCoE, FC, iSCSI, and NFS Protocols

Saad Jafri and Chris Lemmons, NetApp
May 2011 | TR-3916

ABSTRACT

This document compares the performance of 10-GbE FCoE, 10GbE and GbE iSCSI and NFS, and 8Gb FC to 4Gb FC using VMware® vSphere™ 4.1 on NetApp® storage systems. This document compares the individual protocol performance and CPU use at varying workloads.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Both NetApp storage arrays and VMware vSphere natively support a number of data access protocols, including Fibre Channel over Ethernet (FCoE), Fibre Channel (FC), Internet Small Computer System Interface (iSCSI), and Network File System (NFS). Because of the deployment and management differences within each protocol, determining which of these three protocols to use is one of the key steps in designing a virtualized infrastructure. Knowing how each protocol performs in terms of throughput and CPU utilization can be helpful in making decisions about this important design consideration.

NetApp fabric-attached storage (FAS) and V-Series storage systems employ the NetApp Unified Storage Architecture to simultaneously support all major storage protocols—NFS, Common Internet File System protocol (CIFS), iSCSI, FC, and FCoE—in the same storage system. Our unified storage solutions offer the benefits of simplified data management, investment protection, and reduced total cost of ownership.

NetApp extends its leadership in Ethernet storage to deliver increased storage flexibility and efficiency with the introduction of NetApp Unified Connect. Unified Connect leverages the NetApp Unified Storage Architecture to support transmission of NAS, iSCSI, and FCoE storage traffic simultaneously over shared 10-Gigabit Ethernet (10GbE) ports using the NetApp unified target adapter (UTA), commonly referred to as a converged network adapter (CNA). Unified Connect significantly enhances the value proposition of the converged network by offering end-to-end network convergence, from host to storage, in conjunction with FCoE-enabled host adapters and 10GbE switches. NetApp Unified Connect: Taking Convergence to the Next Level provides more information on Unified Connect.

The findings in this technical report were completed by VMware and NetApp. The testing conducted focused on comparing the performance of the following storage protocols in a vSphere 4.1 environment using the NetApp Unified Storage Architecture combined with NetApp Unified Connect.

- FC using 4Gb connectivity
- FC using 8Gb connectivity
- FCoE using 10Gb connectivity
- iSCSI using 1GbE connectivity
- iSCSI using 10GbE connectivity
- NFS using 1GbE connectivity
- NFS using 10GbE connectivity

The results compare the performance of these protocols using realistic workloads with a goal of helping customers make decisions as they build out their virtual infrastructures into the converged data centers of the future.

2 EXECUTIVE SUMMARY

The goal of these tests was to investigate the performance capabilities of the different network protocols supported by both VMware and NetApp, using realistic workloads. These tests were not designed to demonstrate the potential maximum bandwidth available for each protocol, but to simulate real-world environments using low, medium, and high levels of throughput.

During these tests, we used FCoE, FC using both 4Gb and 8Gb connectivity, iSCSI using both 1GbE and 10GbE, and NFS using both 1GbE and 10GbE connectivity. As mentioned, all of the protocols tested used the NetApp Unified Connect architecture, which uses a single NetApp UTA in the storage controllers that handle all of these protocols, individually and collectively, using a single connection through a Cisco® Nexus® 5020 switch. This allows NetApp customers to greatly simplify their data center networks by eliminating the need to add different network adapters to their storage controllers to support multiple protocols.

For these tests, we created a cluster consisting of eight VMware vSphere 4.1 systems running on Fujitsu Primergy RX200 S6 servers configured with two quad-core Intel® Xeon® E5507 Series processors and 48GB of RAM. We used two NetApp FAS6210 controllers in a high-availability (HA) pair configuration to provide storage for the data accessed by the virtual machines (VMs) during the performance testing. The FAS6200 series is an enterprise-class storage system and has the performance and scale to take on the most demanding storage challenges. These systems are built on the industry-leading NetApp Unified Storage Architecture for ease of management, increased efficiencies, and reduced storage costs.

To access the storage, we connected each Fujitsu RX200 S6 server and FAS6210 controller to a pair of Cisco Nexus 5020 switches. Depending on the protocol being tested, each server contained either a dual-port QLogic 8 or 4Gb FC host bus adapter (HBA), a dual-port QLogic 10Gb CAN, or an Intel 1Gb network interface card (NIC). Each FAS6210 controller contained a dual-port 10Gb UTA, regardless of the protocol being tested. The dual ports on each card were split between the two switches so that port 1 was connected to one switch and port 2 was connected to the other. Figure 12 shows the network diagram of the test environment.

For these tests we looked at two different use cases/workloads that we believe are representative of how vSphere data centers currently use NetApp unified storage.

2.1 LARGE NUMBERS OF VIRTUAL MACHINES ACCESSING SHARED DATASTORES

In this use case, we configured a total of 128 VMs spread evenly across the 8 vSphere 4.1 hosts, all accessing a single shared datastore provisioned on the FAS6210A storage controller. During testing, each of the VMs generated a relatively light load consisting of a mix of 75% reads and 25% writes using a fully random access pattern with 4K and 8K request size. While each VM generated relatively low levels of input/output operations per second (IOPS), taken together they generated a significant load against the FAS6210A storage controllers.

In this case, we believe this workload using a 4K request size represents VMs running a variety of general applications utilizing default block sizes seen with typical file systems, including NTFS and EXT3. Using this workload with an 8K request size would potentially represent multiple database applications like Oracle® and Microsoft® SQL Server® running relatively light loads in a consolidated database environment in which performance requirements can be satisfied by sharing storage resources with other applications.

Once the environment was set up, we used the industry standard Iometer benchmark to measure the performance of each protocol using workloads ranging from light to heavy amounts of input/output (I/O) from the vSphere hosts to the NetApp FAS6210 storage used for the tests. The tests were executed with different numbers of VMs ranging from 32, 64, 96, and 128 executing 64, 128, 192, and 256 total outstanding I/Os, respectively. This allowed us to measure the total throughput generated using each protocol at a variety of points simulating low, typical, and heavy workloads as experienced by the vSphere environment. While a typical vSphere environment might not be driven to these levels, it is valuable to know how the protocols behave at extremely high levels of activity.

We compared the test results of all protocols tested to the performance generated using 4Gb FC and found the following items of interest:

- At all load points tested, we found that all of the protocols under test delivered performance within 5% of that delivered by 4Gb FC.
- At all load points tested, we observed that latencies for all protocols under test were within 5% of those delivered by 4Gb FC.
- Unlike previous testing we have conducted, we found that the differences in vSphere server CPU resources consumed by the different protocols during the shared datastore testing were small enough that we believe they no longer warrant a great deal of discussion. Although we did observe differences in vSphere CPU utilization between the protocols, it was generally in the range of 3 percentage points or less.

2.2 SINGLE VM GENERATING HIGH LEVELS OF CONCURRENT I/O USING A NONSHARED DATASTORE

In this use case, we configured a single VM accessing a datastore that was not shared with any other VMs. During testing, this VM generated a total of 128 concurrent I/Os consisting of a mix of 60% reads and 40% writes using a fully random access pattern with an 8K request size. We believe this represents an environment in which I/O-intensive applications like OLTP databases are hosted in a virtualized environment. Note that 8K is the default block size of Oracle and Microsoft SQL database servers.

The VMware virtual platform provides to guest operating systems virtualized versions of hardware storage adapters from BusLogic and LSI Logic that emulate the physical storage adapters on which they are based. The advantage of this emulation is that most operating systems ship drivers for these devices. However, this design precludes the use of performance optimizations that are possible in virtualized environments.

VMware vSphere 4.1 offers the Paravirtual SCSI (PVSCSI) driver that is designed to provide better throughput and lower CPU utilization in virtualized environments in which guest applications are very I/O intensive. During these tests, we measured performance using both the LSI Logic and the PVSCSI drivers to determine the benefits of using the PVSCSI driver in this I/O-intensive environment.

For these tests, we compared the performance generated with each protocol using the PVSCSI driver to that obtained using the LSI Logic driver. Additionally, we compared the VM CPU utilization using each protocol with both the LSI Logic and the PVSCSI drivers. We noted the following:

- We found nearly identical performance in terms of IOPS when using the LSI Logic and the PVSCSI drivers, indicating that both are capable of delivering enterprise-class performance in an I/O-intensive environment.
- Using the PVSCSI driver allowed us to generate nearly identical performance compared to the LSI Logic driver while consuming 20% to 35% fewer VM CPU resources compared to LSI Logic.

We believe these tests demonstrate that the combination of vSphere 4.1 and NetApp unified storage provides enterprise-class performance in a variety of typical production scenarios with any of the protocols supported by VMware and NetApp. The large number of protocols supported clearly provides outstanding flexibility for our customers to move forward with emerging data center standards like Fibre Channel over Ethernet (FCoE) while maintaining the viability of their existing vSphere environments.

The remainder of this report provides the full details of the test results as well as the details of the test configurations and the methodology used to conduct the tests.

3 SHARED DATASTORES, 75/25 RANDOM READ/WRITE WORKLOAD WITH A 4K BLOCK SIZE

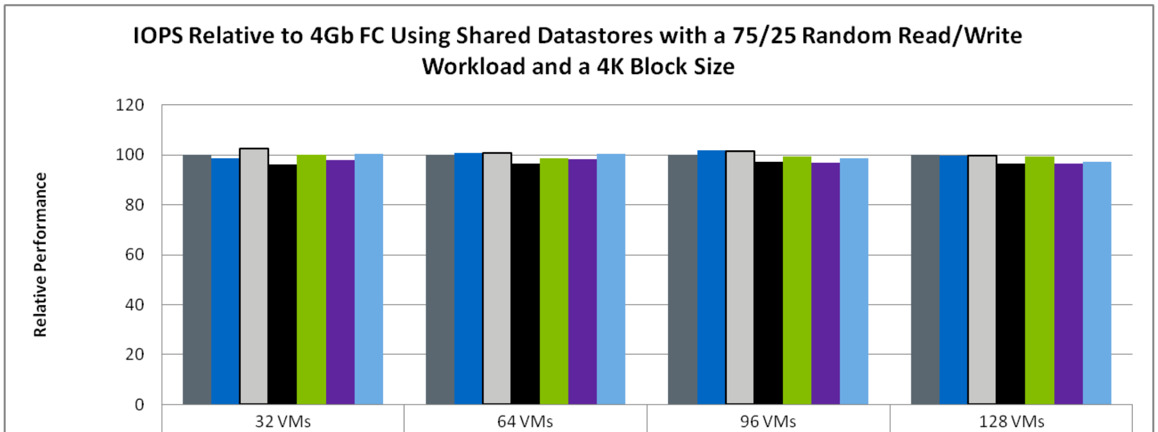
This section provides the specific details of the performance tests for all protocols using shared datastores and a 75/25 random read/write ratio workload with a 4K request size. The tests were executed with different numbers of VMs ranging from 32, 64, 96, and 128 executing 64, 128, 192, and 256 total outstanding I/Os, respectively.

These results are presented in a relative fashion with each protocol compared to the results generated using 4Gb FC as the baseline. For all of these results, throughput and latency results for 4Gb FC are presented in the charts using a value of 100. The throughput and latency results for the remaining protocols under test are then presented relative to 4Gb FC. For example, a throughput result for a specific protocol of 105 indicates that the observed throughput was 5% greater than that observed using 4Gb FC. On the other hand, a throughput result for a specific protocol of 95 indicates that the observed throughput was 5% lower than that observed using 4Gb FC.

3.1 RELATIVE THROUGHPUT COMPARISON

Figure 1 compares the performance of the tested protocols relative to 4Gb FC. We found that all protocols generated performance within 4% of 4Gb FC at low (32 and 64 VMs), medium (96 VMs), and high (128 VMs) loads, indicating excellent performance from all protocols tested.

Figure 1) vSphere throughput for all protocols relative to 4Gb FC with 4K block size.

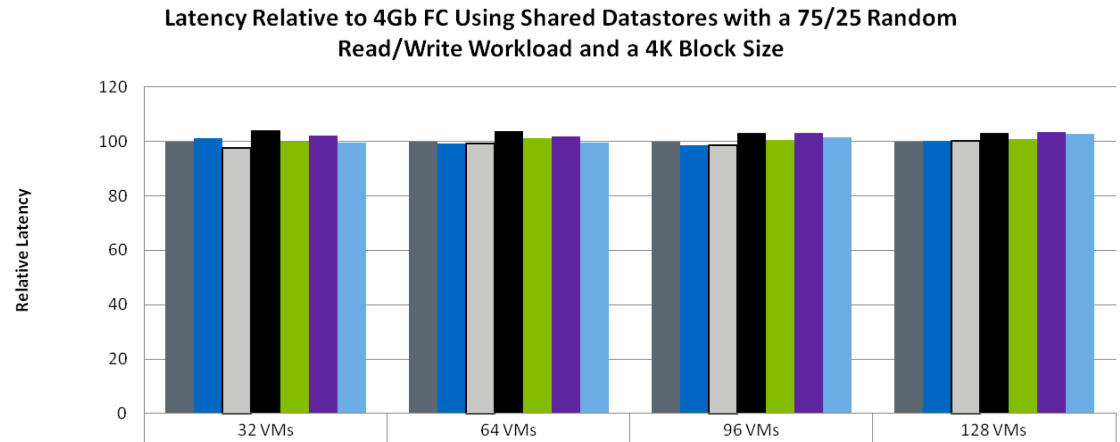


	32 VMs	64 VMs	96 VMs	128 VMs
FC - 4Gb	100	100	100	100
FC - 8Gb	99	101	102	100
FCoE - 10Gb	102	101	101	100
iSCSI - 1GbE	96	97	97	97
iSCSI - 10GbE	100	99	99	99
NFS - 1GbE	98	98	97	97
NFS - 10GbE	100	101	99	97

3.2 RELATIVE LATENCY COMPARISON

Figure 2 compares the latency of the protocols listed relative to 4Gb FC. As expected, we observed average latencies that were within 4% of the latencies observed using 4Gb FC.

Figure 2) vSphere latency for all protocols relative to 4Gb FC with 4K block size.



	32 VMs	64 VMs	96 VMs	128 VMs
FC - 4Gb	100	100	100	100
FC - 8Gb	101	99	99	100
FCoE - 10Gb	98	99	99	100
iSCSI - 1GbE	104	104	103	103
iSCSI - 10GbE	100	101	101	101
NFS - 1GbE	102	102	103	103
NFS - 10GbE	100	99	102	103

4 SHARED DATASTORES, 75/25 RANDOM READ/WRITE WORKLOAD WITH AN 8K BLOCK SIZE

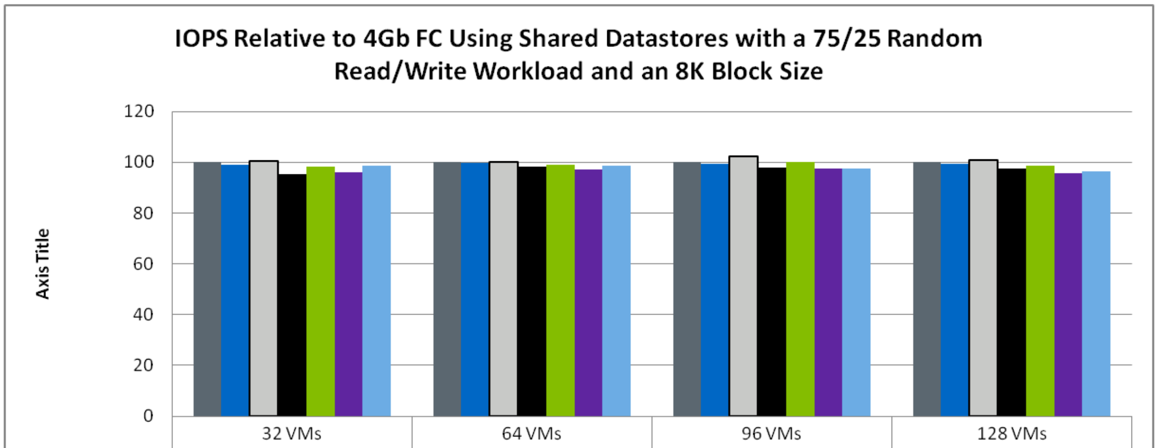
This section provides the specific details of the performance tests for all protocols using shared datastores and a 75/25 random read/write ratio workload with an 8K request size. The tests were executed with different numbers of VMs ranging from 32, 64, 96, and 128 executing 64, 128, 192, and 256 total outstanding I/Os, respectively.

These results are presented in a relative fashion with each protocol compared to the results generated using 4Gb FC as the baseline. For all of these results, throughput and latency results for 4Gb FC are presented in the charts using a value of 100. The throughput and latency results for the remaining protocols under test are then presented relative to 4Gb FC. For example, a throughput result for a specific protocol of 105 indicates that the observed throughput was 5% greater than that observed using 4Gb FC. On the other hand, a throughput result for a specific protocol of 95 indicates that the observed throughput was 5% lower than that observed using 4Gb FC.

4.1 RELATIVE THROUGHPUT COMPARISON

Figure 3 compares the performance of the tested protocols relative to 4Gb FC. We found that all protocols generated performance within 5% of 4Gb FC at low (32 and 64 VMs), medium (96 VMs), and high (128 VMs) loads, indicating excellent performance from all protocols tested.

Figure 3) vSphere throughput for all protocols relative to 4Gb FC with 8K block size.

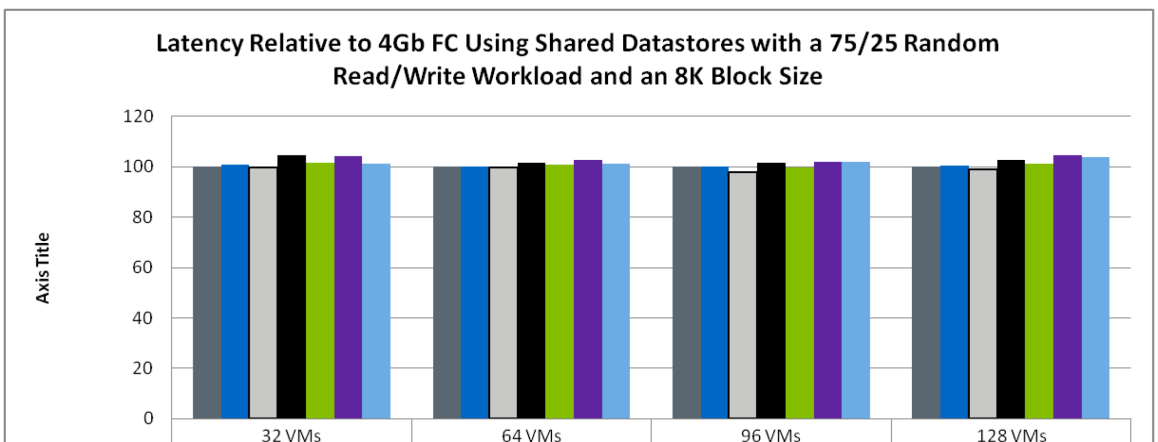


	32 VMs	64 VMs	96 VMs	128 VMs
FC - 4Gb	100	100	100	100
FC - 8Gb	99	100	99	99
FCoE - 10Gb	100	100	102	101
iSCSI - 1GbE	95	98	98	97
iSCSI - 10GbE	98	99	100	99
NFS - 1GbE	96	97	98	96
NFS - 10GbE	99	99	98	96

4.2 RELATIVE LATENCY COMPARISON

Figure 4 compares the latency of the tested protocols relative to 4Gb FC. As expected, we observed average latencies that were within 4% of the latencies observed using 4Gb FC.

Figure 4) vSphere latency for all protocols relative to 4Gb FC with 8K block size.



	32 VMs	64 VMs	96 VMs	128 VMs
FC - 4Gb	100	100	100	100
FC - 8Gb	101	100	100	101
FCoE - 10Gb	100	100	98	99
iSCSI - 1GbE	105	102	102	103
iSCSI - 10GbE	102	101	100	101
NFS - 1GbE	104	103	102	105
NFS - 10GbE	101	101	102	104

5 HIGH-PERFORMANCE NONSHARED DATASTORE, 60/40 RANDOM READ/WRITE WORKLOAD WITH AN 8K BLOCK SIZE

This section provides the specific details of the performance tests for all protocols using nonshared datastores and a workload consisting of a 60/40 random read/write mix using an 8K block size generated from a single VM using both LSI Logic and PVSCSI drivers. This testbed was designed to provide some guidance on what to expect with the default block size used with Oracle and Microsoft SQL database servers.

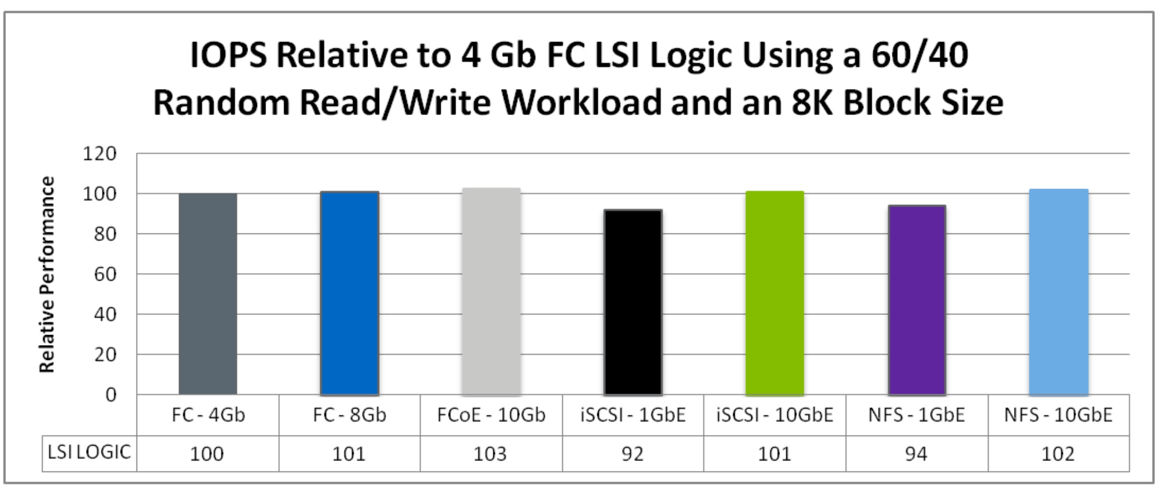
Two sets of data are reported for these tests. We compared the performance of each protocol relative to 4Gb FC, first using only the LSI Logic driver and then using only the PVSCSI driver. These results are presented in a relative fashion with each protocol compared to the results generated using 4Gb FC as the baseline. For all these results, throughput and latency results for 4Gb FC using either the LSI Logic or PVSCSI driver are presented in the charts using a value of 100. The throughput and latency results for the remaining protocols under test are then presented relative to 4Gb FC. For example, a throughput result for a specific protocol using the LSI Logic driver of 105 indicates that the observed throughput was 5% greater than that observed using 4Gb FC with the LSI Logic driver. On the other hand, a throughput result for a specific protocol of 95 indicates that the observed throughput was 5% lower than that observed using 4Gb FC with the same disk driver.

Additionally, results are presented in a relative fashion with each protocol compared to the results generated using the LSI Logic driver as the baseline. In the following graphs, throughput, latency, and VM CPU results for each protocol using the LSI Logic driver are presented in the charts using a value of 100. The throughput, latency, and VM CPU utilization for the same protocols using the PVSCSI driver are then presented relative to those observed using the LSI Logic driver. For example, a throughput result for a specific protocol of 105 using the PVSCSI driver indicates that the observed throughput was 5% greater than that observed using the same protocol and the LSI Logic driver.

5.1 RELATIVE THROUGHPUT COMPARISON USING LSI LOGIC

Figure 5 compares the performance of the tested protocols relative to 4Gb FC when using the LSI Logic driver. These results show that all protocols utilizing larger than 1GbE network links generated performance within 3% of 4Gb FC and that protocols using 1GbE network links generated performance within 6% to 8% of 4Gb FC in this I/O-intensive environment.

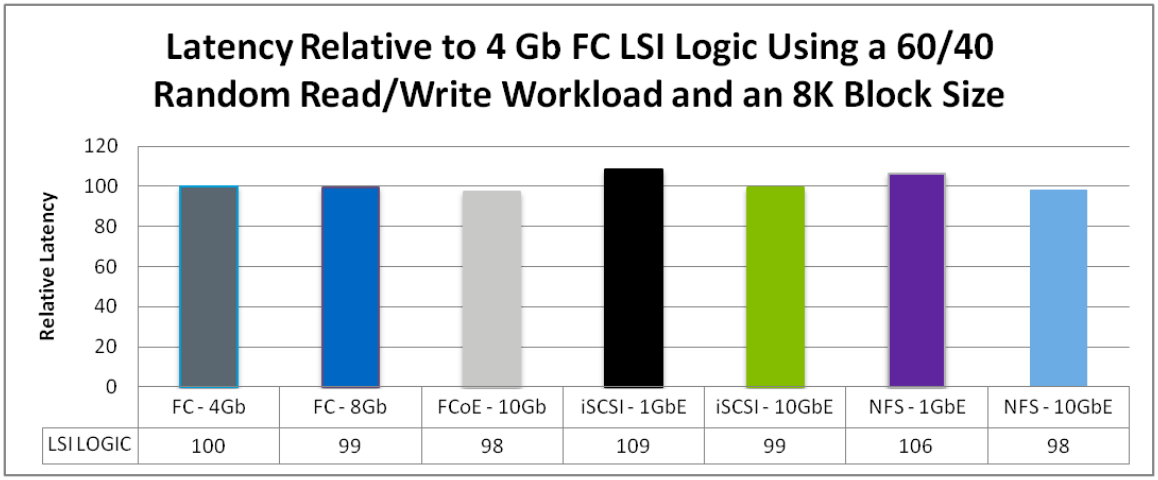
Figure 5) vSphere throughput for all protocols using LSI Logic relative to 4Gb FC with 8K block size.



5.2 RELATIVE LATENCY COMPARISON USING LSI LOGIC

Figure 6 compares the average latencies observed for each of the protocols relative to 4Gb FC using the LSI Logic driver. In this case, we observed parity from a latency perspective similar to what we observed when comparing the throughput generated using the LSI Logic driver.

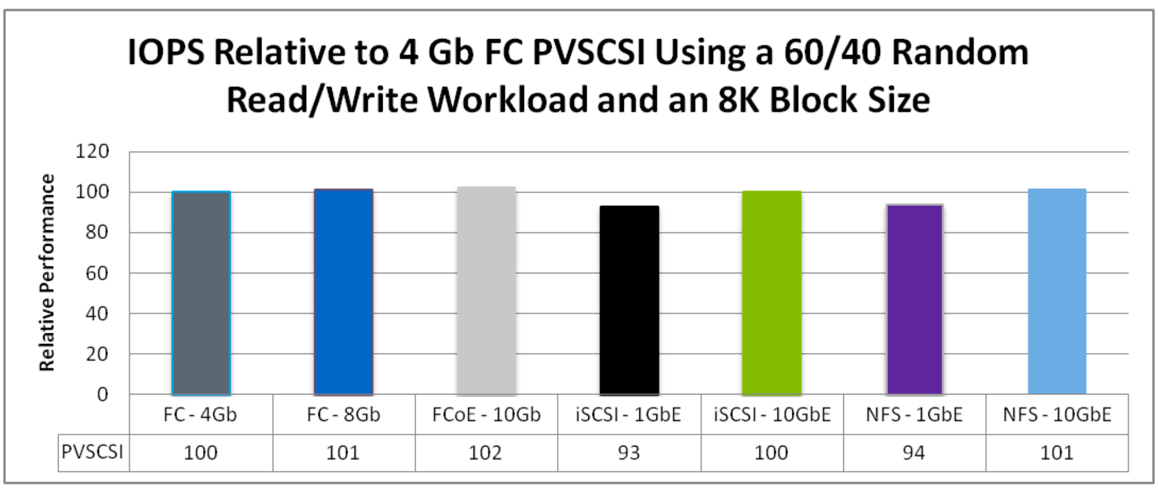
Figure 6) vSphere latency for all protocols using LSI Logic relative to 4Gb FC with 8K block size.



5.3 RELATIVE THROUGHPUT COMPARISON USING PVSCSI

Figure 7 compares the performance of the tested protocols relative to 4Gb FC when using the PVSCSI driver. These results show that all protocols utilizing larger than GbE network links generated performance within 2% of 4Gb FC and that protocols using GbE network links generated performance within 6% to 7% of 4Gb FC in this I/O-intensive environment.

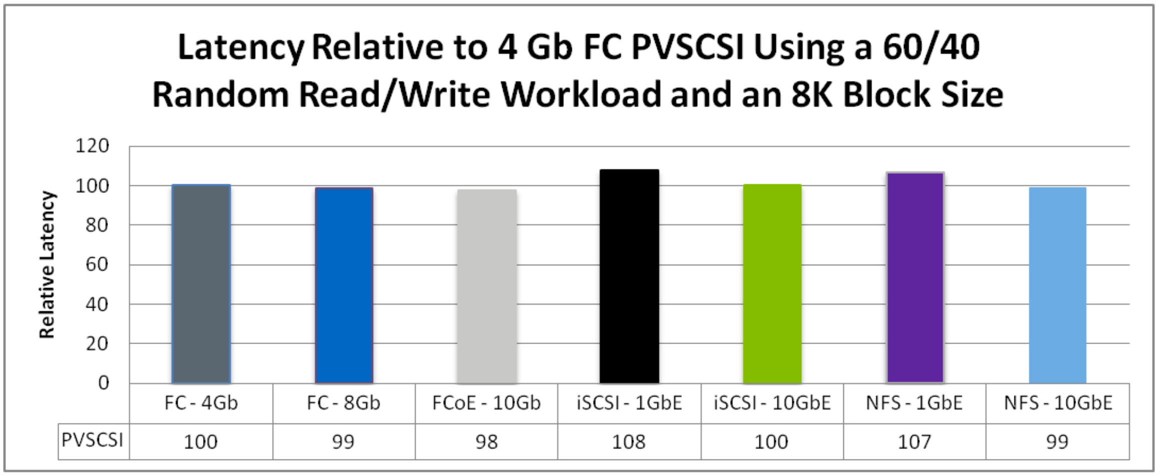
Figure 7) vSphere throughput for all protocols using PVSCSI relative to 4Gb FC with 8K block size.



5.4 RELATIVE LATENCY COMPARISON USING PVSCSI

Figure 8 compares the average latencies observed for each of the protocols relative to 4Gb FC using the PVSCSI driver. In this case, the parity we observed from a latency perspective was similar to what we observed when comparing the throughput generated using the PVSCSI driver.

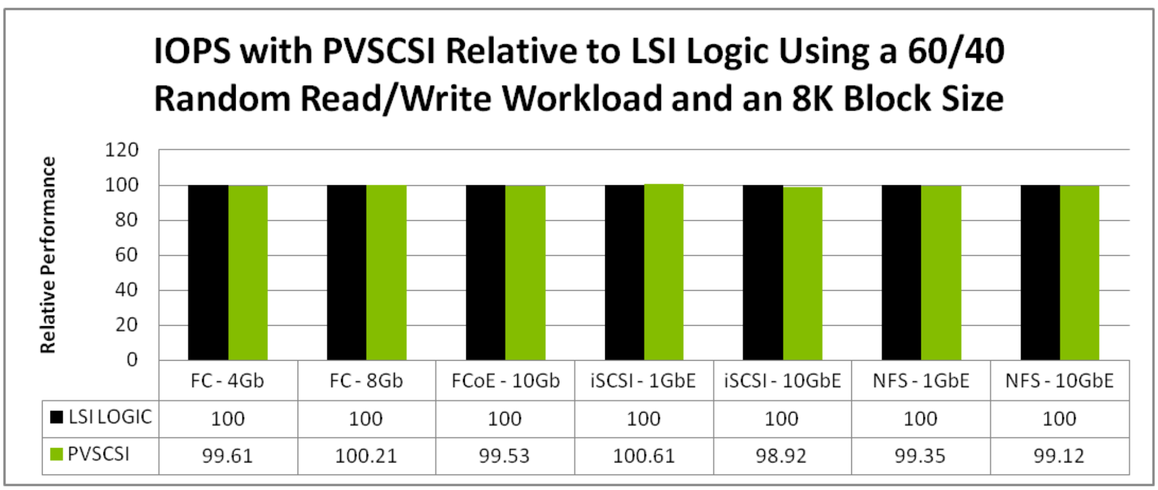
Figure 8) vSphere latency for all protocols using PVSCSI relative to 4Gb FC with 8K block size.



5.5 THROUGHPUT COMPARISON OF INDIVIDUAL PROTOCOLS USING PVSCSI RELATIVE TO LSI LOGIC

Figure 9 compares the throughput generated using both the LSI Logic and the PVSCSI drivers for each of the individual protocols tested. These results show nearly identical throughput capabilities when using either the LSI Logic or the PVSCSI drivers in this I/O-intensive environment.

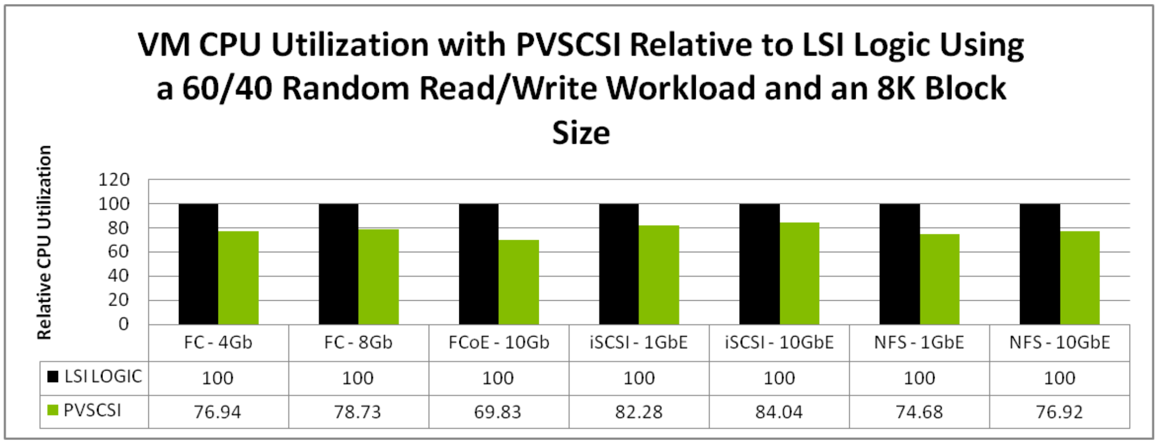
Figure 9) vSphere throughput for PVSCSI relative to LSI Logic with 8K block size for each protocol.



5.6 VM SERVER CPU UTILIZATION COMPARISON OF INDIVIDUAL PROTOCOLS USING PVSCSI RELATIVE TO LSI LOGIC

Figure 10 compares the average CPU utilization observed in the single VM used to drive the load for these tests for each of the individual protocols tested. Depending on the protocols, we found that using the PVSCSI driver required 18% to 30% fewer CPU resources in the VM compared to using the LSI Logic driver. This indicates that the PVSCSI driver delivered the same IOPS much more efficiently than the LSI Logic driver with all protocols tested.

Figure 10) Virtual machine CPU utilization for PVSCSI relative to LSI Logic with 8K block size for each protocol.

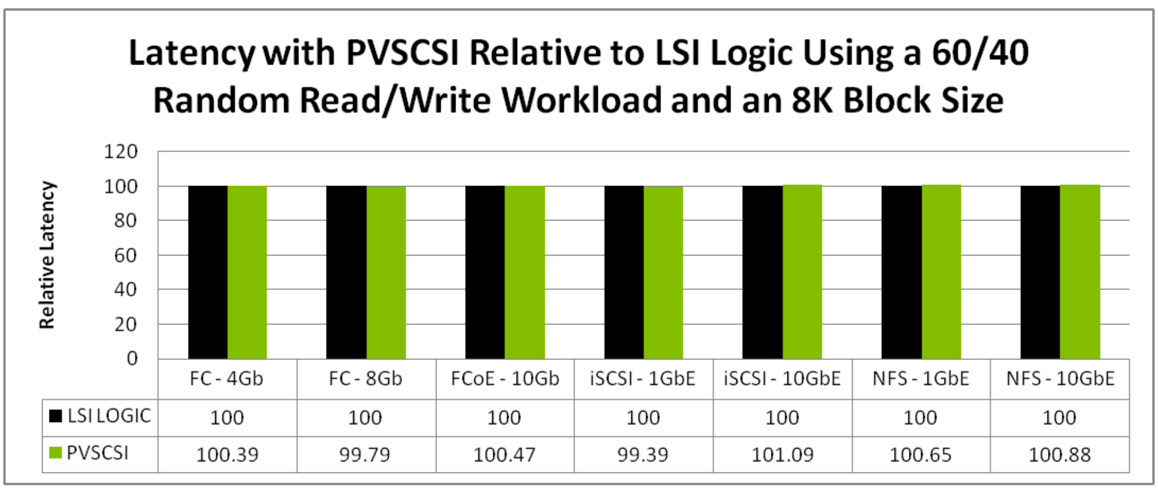


	FC - 4Gb	FC - 8Gb	FCoE - 10Gb	iSCSI - 1GbE	iSCSI - 10GbE	NFS - 1GbE	NFS - 10GbE
LSI LOGIC	100	100	100	100	100	100	100
PVSCSI	76.94	78.73	69.83	82.28	84.04	74.68	76.92

5.7 LATENCY COMPARISON OF INDIVIDUAL PROTOCOLS USING PVSCSI RELATIVE TO LSI LOGIC

Figure 11 compares the average latencies observed using both the LSI Logic and the PVSCSI drivers for each of the individual protocols tested. In this case, the parity we observed from a latency perspective was similar to what we observed when comparing the throughput generated using the LSI Logic and the PVSCSI drivers.

Figure 11) vSphere latency for PVSCSI relative to LSI Logic with 8K block size for each protocol.



	FC - 4Gb	FC - 8Gb	FCoE - 10Gb	iSCSI - 1GbE	iSCSI - 10GbE	NFS - 1GbE	NFS - 10GbE
LSI LOGIC	100	100	100	100	100	100	100
PVSCSI	100.39	99.79	100.47	99.39	101.09	100.65	100.88

6 TEST DESIGN AND CONFIGURATION

This section provides the details of the hardware used for the testing. This includes how the NetApp storage was provisioned and presented to the vSphere 4.1 hosts as well as the infrastructure required to provide the connectivity between the vSphere 4.1 hosts and NetApp storage.

At a high level, we configured the VMware infrastructure with 8 vSphere 4.1 hosts , a vCenter™ 4.1 server, and the NetApp FAS6210A storage platform configured with 192 15K SAS drives and a dual-port 10Gb UTA. Depending on the protocol being tested, each vSphere 4.1 host also contained either a QLogic QLE8152 10Gb CNA and QLogic QLE2562 8Gb FC HBA or a QLogic QLE2462 4Gb FC HBA and an Intel dual-port 1GbE NIC. The network connectivity between the vSphere 4.1 hosts and the

NetApp FAS6210 storage controllers was through two Cisco Nexus 5020 switches supporting FCoE, traditional 8 and 4Gb FC, and 1 and 10GbE.

6.1 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 and Table 2 provide the details of the hardware and software components that we used to create the vSphere 4.1 data center for the testing. The data center consisted of a cluster of eight vSphere 4.1 host servers connected using 1 and 10GbE FCoE and 4 and 8Gb FC connections to the NetApp FAS6210A storage controllers.

When conducting testing, we managed the eight vSphere 4.1 hosts using a vCenter 4.1 server running a vSphere client.

Table 1) Hardware and software components used for vSphere 4.1 data center.

Component	Details
Virtual infrastructure	VMware vSphere 4.1 build 320092 VMware vCenter 4.1 build 258902
Servers	8 x Fujitsu Primergy RX200 S6
Processors	2 quad-core Intel Xeon E5507 Series at 2.3GHz
Memory	48GB
Ethernet network	10GbE and 1GbE
Fibre Channel network	8Gb and 4Gb
Host bus adapters	QLogic dual-port PCIe 10GbE CNA - QLE8142 QLogic dual-port PCIe FC 8Gb - QLE2562 QLogic dual-port PCIe FC 4Gb - QLE2462 Intel PRO/1000 PT PCIe 1GbE
Network switches	Cisco Nexus 5020 series

Table 2) Virtual center server details.

Component	Details
Server	1 x Fujitsu Primergy RX200 S6
Processors	1 quad-core Intel Xeon E5504 Series at 2.0GHz
Memory	6GB
Operating system	Microsoft Windows® Server® 2008 R2 Enterprise

Table 3 describes the system used to generate the Iometer workload.

Table 3) Iometer workload generation controller server.

Component	Details
Server	1 x Fujitsu Primergy RX200 S6
Processors	2 quad-core Intel Xeon E5420 Series at 2.5GHz
Memory	34GB

Component	Details
Operating system	Microsoft Windows Server 2008 R2 Enterprise

Table 4 describes the NetApp products used to provide the shared storage for the test configuration.

Table 4) NetApp storage hardware and software components.

Component	Details
Storage system	1 x FAS6210A
Data ONTAP® version	8.0.1RC2 (7-Mode)
Number of drives	192
Size of drives	300GB
Speed of drives	15K
Type of drives	SAS

Note: There is approximately 18TB of usable storage on each FAS6210 storage controller.

6.2 CONNECTING THE NETAPP FAS6210 TO THE VMWARE DATA CENTER

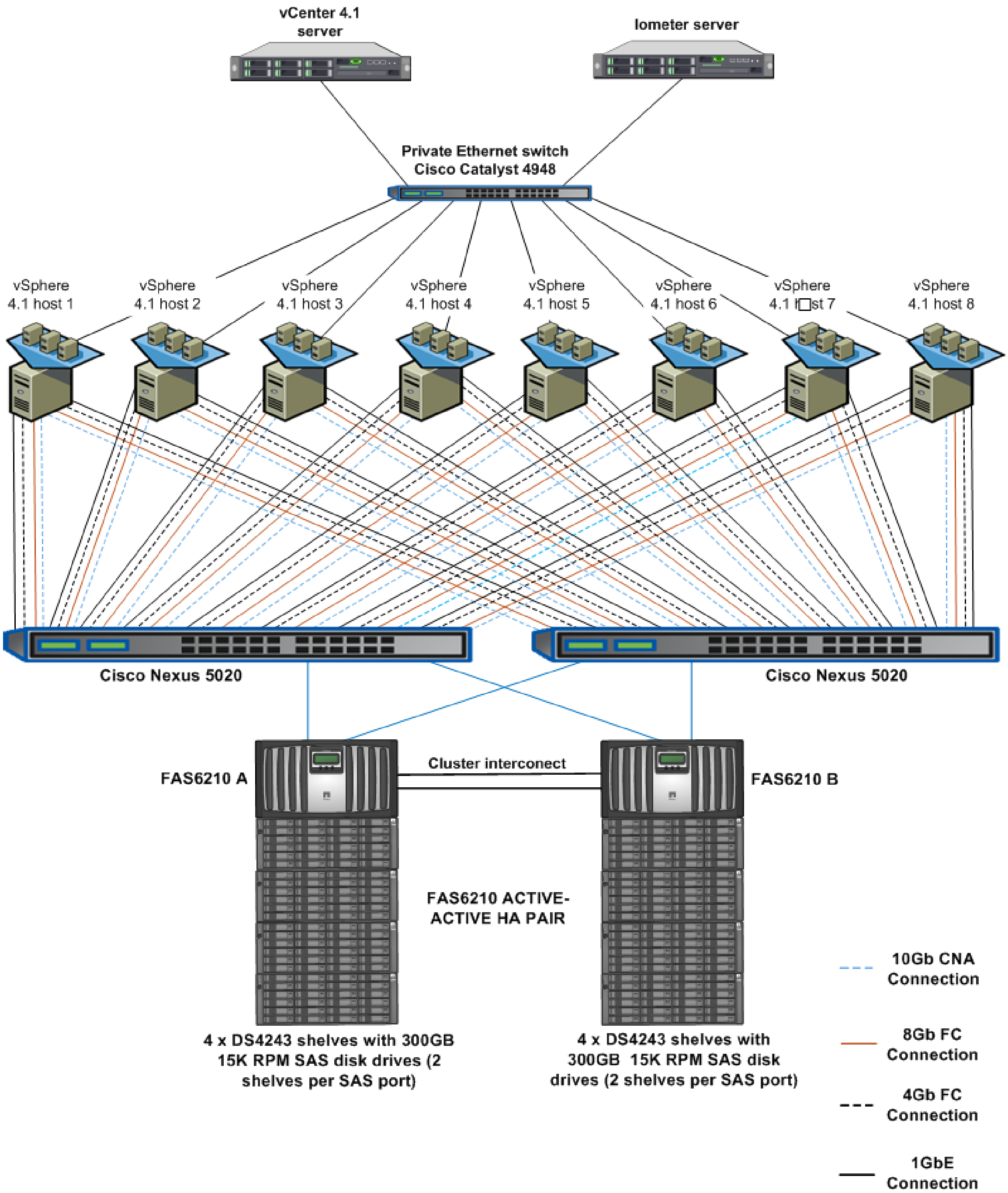
This section provides the details about how the NetApp FAS6210A storage controllers were connected to the VMware data center created for the testing.

The network diagram in Figure 12 shows how the vSphere 4.1 servers and NetApp FAS6210A storage controllers were physically connected for all tests using 4Gb and 8Gb FC, FCoE, 1GbE, and 10GbE using both iSCSI and NFS. Cisco Nexus 5020 switches were used to provide the connectivity, and each vSphere 4.1 server was connected to both switches to provide multiple paths to their respective storage.

The vCenter and Iometer controller servers were connected to the VMware data center through a separate Cisco Catalyst® 4948 Gigabit Ethernet switch.

For all performance tests, the FAS6210A storage controllers were configured with a dual-port 10Gb UTA that handled all the different protocols tested using a single connection into the storage controller. This greatly simplified the process of running multiple protocols in this unified storage environment.

Figure 12) FC, FCoE, iSCSI, and NFS connectivity between vSphere 4.1 hosts and NetApp storage.



6.3 PROVISIONING NETAPP STORAGE TO THE VMWARE DATA CENTER FOR TESTING USING FCOE, FC, AND ISCSI

For testing the block-based protocols, each of the FAS6210 storage controllers was configured with a total of 96 300GB/15K RPM SAS disk drives. During these tests, we created a single set of logical unit numbers (LUNs) on the FAS6210A storage controllers for use by all of the block-based protocols (FCoE, FC, and iSCSI) under test. Switching between protocols was then as simple as mapping a different igroup to the existing LUNs and rescanning the storage from the vSphere 4.1 hosts.

Two separate aggregates were created on each of the FAS6210 storage controllers. One was the root aggregate, containing three disk drives to house the Data ONTAP software. The second aggregate contained 92 disk drives configured into four RAID-DP® groups of 23 disks (21+2) each to provide shared storage for the VMs created for testing. Additionally, we maintained a spare disk on each controller.

After the aggregates were created as described, a set of FlexVol® volumes and LUNs were created on each of the FAS6210 storage controllers to be presented to the vSphere 4.1 hosts as follows:

- FlexVol volume and Virtual Machine File System (VMFS)-formatted LUN containing the VM's operating system (OS) virtual disks used when running any of the block-based protocols under test
- FlexVol volume and VMFS-formatted LUNs containing the VM's virtual disks used to hold Iometer workload data files used by any of the block-based protocols under test

Table 5 summarizes the aggregates, FlexVol volumes, and LUN layout for testing using block-based protocols.

Table 5) FA6210A aggregate, FlexVol volumes, and LUNs for testing using block-based protocols.

NetApp Storage Controller	Aggregate Name	Volume Name	LUN	Size	Description
Controller 1	aggr0			3 disks	
		vol0		227GB	Root volume
	aggr1			92disks	
		blkboot_c1vol1		4TB	
			blkboot_c1lun1	2TB	VM OS disks
		blkdata_c1vol1		4TB	
			blkdata_c1lun1	2TB	Iometer data files
Controller 2	aggr0			3 disks	
		vol0		227GB	Root volume
	aggr1			92disks	
		blkboot_c2vol2		4TB	
			blkboot_c2lun2	2TB	VM OS disks
		blkdata_c2vol2		4TB	
			blkdata_c2lun2	2TB	Iometer data files

After the creation of FlexVol volumes and their respective LUNs was completed, as shown in Table 5, four different initiator groups were created on each NetApp FAS6210 storage controller in order to present the LUNs to vSphere 4.1 hosts as follows:

- FC igroup containing WWPNs of the 10Gb CNAs in the vSphere 4.1 hosts
- FC igroup containing WWPNs of 8Gb FC HBAs in the vSphere 4.1 hosts
- FC igroup containing WWPNs of 4Gb FC HBAs in the vSphere 4.1 hosts
- iSCSI igroup containing the IQNs of the software iSCSI initiator in the vSphere 4.1 hosts

After the initiator groups were created using information from the vSphere 4.1 hosts, the initiator groups were mapped to their respective LUNs, depending on which of the block protocols were being tested.

Once the LUNs were mapped to the proper initiator group, the storage was available to the vSphere 4.1 hosts in the cluster.

6.4 PROVISIONING NETAPP STORAGE TO THE VMWARE DATA CENTER FOR TESTING USING NFS

After all tests using any of the block-based protocols were completed, the same set of tests was conducted using the NFS protocol. For testing using NFS, we deleted all of the aggregates, FlexVol volumes, and VMFS-formatted LUNs on the FAS6210A storage controllers used to store the VM boot data and Iometer data for the block protocol tests. We did not delete the aggregate and FlexVol volume used to store the Data ONTAP software.

We then created a new aggregate containing 92 disk drives configured into 4 RAID-DP groups of 23 disks (21+2) each to provide shared storage for the VMs created for testing using NFS. We also maintained a spare disk on each controller.

This configuration provided approximately 32TB of usable shared storage (18TB per storage controller). After the aggregates were completed as described, a set of FlexVol volumes was created on each of the FAS6210 storage controllers to be presented to the vSphere 4.1 servers as follows:

- FlexVol volume containing the VM's OS virtual disks used when running NFS
- FlexVol volume containing the VM's virtual disks used to hold Iometer workload data files used during the NFS testing

Table 6 summarizes the aggregates/volumes layout for testing using the NFS protocol.

Table 6) FA6210A aggregate and FlexVol volumes for testing using NFS protocol.

NetApp Storage Controller	Aggregate Name	Volume Name	Size	Description
Controller 1	aggr0		3 disks	
		vol0	227GB	Root volume
	aggr1		92 disks	
		nfsboot_c1vol1	2TB	VM OS disks
		nfsdata_c1vol1	2TB	Iometer data files
Controller 2	aggr0		3 disks	
		vol0	227GB	Root volume
	aggr1		92 disks	
		nfsboot_c2vol2	2TB	VM OS disks
		nfsdata_c2vol2	2TB	Iometer data files

6.5 CONFIGURING THE VMWARE VSPHERE DATA CENTER

After provisioning the storage as described in the previous sections, we created 16 VMs on each of the 8 vSphere 4.1 hosts for use during the tests. As mentioned, the eight vSphere 4.1 hosts are identical Fujitsu Primergy RX200 S6 servers, each configured with 48GB of RAM and two quad-core Intel Xeon E55507 Series processors. Table 7 provides the configuration details for each of the VMs.

Table 7) Virtual machine components.

Component	Details
Operating System	Windows Server 2008 R2
Number of Virtual Processors	1
Memory	1GB
Virtual Disk Size for Iometer Data	10GB
Virtual Disk Size for OS	10GB

To use both FAS6210 storage controllers for these tests and have a balanced load during tests, we distributed the 16 VMs on each vSphere 4.1 host equally across both storage controllers. On each vSphere 4.1 host, we configured the OS virtual disk (drive C:\) for VMs 1 through 8 to reside in an aggregate on one NetApp storage controller and VMs 9 through 16 to reside in an aggregate on the other NetApp storage controller.

We also configured each VM with two additional virtual disks (drives D:\ and E:\) to contain the Iometer test data files accessed during the testing. For each VM, one of these virtual drives (drive D:\) was placed in an aggregate on one storage controller, and the other virtual drive (drive E:\) was placed in an aggregate on the other storage controller. This enabled each of the VMs participating in a test to read and write data to a single VMware VMFS or NFS datastore on both FAS6210 storage controllers. The idea behind this was to have larger numbers of VMs accessing fewer shared datastores to better simulate real-world scenarios.

To create the 128 VMs for the testing, we created a gold master VM as follows:

1. Created a VM in a mounted datastore of vSphere 4.1 server 1. Used the datastore designated to contain the VM's OS virtual drive to store VM files.
2. Before installing the operating system, followed TR-3747: Best Practices for File System Alignment in Virtual Environments.
3. Installed the Windows Server 2008 R2 OS on the VM using the virtual disk created in step 1. This virtual disk was the VM OS virtual disk with drive letter C:\ assigned.
4. Configured the network adapter on the VM to use Dynamic Host Configuration Protocol (DHCP) to retrieve an IP address.
5. Installed the Iometer Dynamo application that would be used later to generate the workload for all test cases.

After creating the gold master VM, we then created two 10GB virtual disks in the gold master VM for use by Iometer so that the workload generated for each protocol under test was evenly distributed across both of the FAS6210 storage controllers. As shown in the previous section, each of the eight vSphere 4.1 hosts has access to two shared datastores for each protocol under test with one of the datastores provisioned on the first FAS6210 storage controller and the second datastore provisioned on the second FAS6210 storage controller. When creating the two virtual disks, we placed one virtual disk in the datastore provisioned on the first FAS6210 storage controller and the second virtual disk on the second FAS6210 storage controller.

After creating the gold master and the two virtual disks designated for Iometer use, we used standard VMware cloning through the vSphere client running on a vCenter 4.1 server to clone 128 VMs (16 VMs per vSphere 4.1 host) spread evenly across the 2 NetApp FAS6210 storage controllers. Each of the cloned VMs was then customized using a vCenter 4.1 server customization specification to enable the cloned VMs to have unique computer names and SSIDs. Once they were powered on, the VMs received unique IP addresses from a standalone DHCP server.

Table 8 shows the breakdown of how the odd- and even-numbered VMs on one of the eight vSphere 4.1 hosts were spread across both storage controllers with their designated volumes/LUNs and Iometer drive letters for any of the block-based protocols under test. This layout was applied to each vSphere 4.1 host, resulting in all eight nodes in the cluster sharing the same datastore for any of the block-based protocols under test.

Table 8) Virtual machine layouts and their respective virtual disk drive letters across the FAS6210A storage controllers for any of the block protocols under test.

VM #	vSphere Host	Virtual Disk Label and Drive Letter	NetApp Storage Controller	Volume	LUN
1, 3, 5, 7, 9, 11, 13, 15	1	OS drive C:\	Controller 1	blkboot_c1vol1	blkboot_c1lun1
		iomblk_c1l1 D:\	Controller 1	blkdata_c1vol1	blkdata_c1lun1
		iomblk_c2l2 E:\	Controller 2	blkdata_c2vol2	blkdata_c2lun2
2, 4, 6, 8, 10, 12, 14, 16	1	OS drive C:\	Controller 2	blkboot_c2vol2	blkboot_c2lun2
		iomblk_c1l1 D:\	Controller 1	blkdata_c1vol1	blkdata_c1lun1
		iomblk_c2l2 E:\	Controller 2	blkdata_c2vol2	blkdata_c2lun2

Table 9 shows the breakdown of how the odd- and even-numbered VMs on one of the eight vSphere 4.1 hosts were spread across both storage controllers with their designated volumes and Iometer drive letters for the NFS protocol under test. This layout was applied to each vSphere 4.1 host, resulting in all eight nodes in the cluster sharing the same datastore for NFS.

Table 9) Virtual machine layouts and their respective virtual disk drive letters across the FAS6210A storage controllers for NFS protocol.

VM #	vSphere Host	Virtual Disk Label and Drive Letter	NetApp Storage Controller	Volume
1, 3, 5, 7, 9, 11, 13, 15	1	OS drive C:\	Controller 1	nfsboot_c1vol1
		iomnfs_c1v1 D:\	Controller 1	nfsdata_c1vol1
		iomnfs_c2v2 E:\	Controller 2	nfsdata_c2vol2
2, 4, 6, 8, 10, 12, 14, 16	1	OS drive C:\	Controller 2	nfsboot_c2vol2
		iomnfs_c1v1 D:\	Controller 1	nfsdata_c1vol1
		iomnfs_c2v2 E:\	Controller 2	nfsdata_c2vol2

In addition to the configurations discussed earlier, the following additional changes were made to all of the vSphere 4.1 hosts:

- Set FC HBA queue depth to 255.

- Set iSCSI queue depth to 255.

We also set the following option on the NetApp storage controllers. This option changes the runtime priority of the process that handles the SCSI protocol for incoming Fibre Channel requests.

- **`Options wafl_downgrade_target off`**

For all tests with iSCSI and NFS, jumbo frames were enabled by issuing the following commands on each of the vSphere 4.1 hosts, Cisco Nexus 5020 switches, and NetApp storage controllers.

To enable jumbo frames on vSphere 4.1 servers, the following commands were issued on each vSphere 4.1 host:

- **`esxcfg-vmknic -d VMkernel`**
- **`esxcfg-vmknic -a -i <IP addr> -n <netmask> -m 9000 VMkernel`**
- **`esxcfg-vswitch -m 9000 <vSwitch name>`**

To enable jumbo frames on Cisco Nexus switches, the following commands in configuration mode were issued on each switch:

- **`policy-map jumbo`**
- **`class class-default`**
- **`mtu 9216`**
- **`system qos`**
- **`service-policy jumbo`**

To enable jumbo frames on NetApp storage controllers, the following command was issued on each storage controller:

- **`ifconfig<interface> <IP addr> netmask <netmask addr> mtusize 9000 partner <interface>`**

When iSCSI was tested over 1GbE links, multiple TCP session support was enabled on each vSphere 4.1 host according to TR-3749: NetApp and VMware vSphere Storage Best Practices. For FCoE, iSCSI, and FC tests, the round-robin load-balancing policy was enabled for respective datastores on each vSphere 4.1 host.

6.6 WORKLOAD DEFINITION AND IOMETER ENVIRONMENT

For the protocol performance testing, the publicly available Iometer (found at www.iometer.org) application was used to generate the load. Iometer is a client-server application that works as both a workload generator and a measurement tool. The client portion is called “Dynamo” and is installed on each of the VMs. The server portion is called the “Iometer controller” and is installed on a standalone server separate from the vSphere 4.1 hosts. The Iometer controller is used to manage the Dynamo load generators running on the VMs and to gather test results from each of the Dynamos.

A dedicated server running Microsoft Windows Server 2008 R2 was configured to serve as both a Domain Name System (DNS) server to the VMs and an Iometer controller. A separate dedicated server running Microsoft Windows 2008 R2 was configured as an active directory and DHCP server to the VMs and also served as the vCenter 4.1 server.

LARGE NUMBERS OF VIRTUAL MACHINES ACCESSING SHARED DATASTORES

The workloads used for VMs accessing shared datastores were a mixture of random reads and writes using both 4K and 8K request sizes, which represent realistic workloads experienced by vSphere 4.1 hosts in production environments. The specifics of these loads are as follows:

- 4K request size, 75% read, 25% write, 100% random
- 8K request size, 75% read, 25% write, 100% random

After the VM cloning was completed as described, we used Iometer to initialize the virtual disks in each VM with 10GB Iometer data files that would be used during the performance tests. The tests involved all 128 VMs.

We executed the following steps to initialize Iometer target disks:

1. Powered on all 128 VMs.
2. Ran the Iometer Dynamo application in each of the 128 VMs.
3. Launched the Iometer controller application on the server designated as the Iometer controller server.
4. In the Iometer controller application, selected both drive letters corresponding to the Iometer target disks for all 128 VMs.
5. Created an Iometer access specification to generate a 100% random-read load using 4K request size.
6. Executed the specification on all 128 VMs for a total of 5 minutes with no ramp-up time.

The results of the random-read test runs were discarded; the test served only to create a 10GB Iometer data file in the virtual disks associated with the VMs. The process of initializing virtual disks created Iometer data files (named `iobw.tst`) in each of the two disk targets (D:\ and E:\) that were seen by the Iometer controller application and that would be used by it during the testing.

After initializing the virtual disks seen by the Iometer controller, we powered off all 128 VMs and created a Snapshot® copy of the aggregate defined in Table 5 and Table 6. This Snapshot copy represented the initial state of the system before any tests were performed that modified the data files. At the end of every test, we powered off all VMs and used this Snapshot copy to restore the storage to the freshly initialized state so that all tests started from exactly the same point in time with the Iometer data files in their initialized state.

For testing with FCoE, iSCSI, and FC, the same set of LUNs was used to hold the VM's OS virtual disks, and another set of LUNs was used to hold the VM's virtual disks containing the Iometer data files. Therefore, in order to change the block protocol under test, we unmapped/remapped the relevant LUNs holding the VM boot OS disks and VMDK files containing the Iometer data files to an initiator group that corresponded to the protocol being tested. We then rescanned for datastores in the vSphere 4.1 client so that the datastores were seen by all eight vSphere 4.1 hosts using the corresponding protocol being tested.

For tests using 128 VMs, an Iometer access specification was configured so that each Dynamo instance running on an individual VM generated a constant total of 2 outstanding I/Os. We then increased the load by increasing the total number of VMs participating in a given test. Additionally, each of the 128 VMs participating in the test accessed 20GB of Iometer data for total working sets ranging from 320GB to 1280GB, depending on the number of VMs participating in the given test.

Table 10 summarizes the four different test configurations used for each different protocol configuration. It includes the number of VMs used, the total number of outstanding I/Os generated by each vSphere host, and the total number of outstanding I/Os generated against the NetApp storage from the entire VMware data center.

Table 10) Workload scaling keeping eight outstanding I/Os per VM.

Number of VMs per vSphere 4.1 Server	Outstanding I/O per vSphere 4.1 Server	Total Outstanding I/Os
4	8	64
8	16	128

Number of VMs per vSphere 4.1 Server	Outstanding I/O per vSphere 4.1 Server	Total Outstanding I/Os
12	24	192
16	32	256

For all test cases, the Iometer tests were run for a total of 15 minutes, with a 10-minute ramp-up time to allow performance to reach a steady state before Iometer began to record results.

For each workload, we measured the following items on vSphere servers:

- Throughput in IOPS of FCoE, iSCSI, NFS, and FC protocols as reported by Iometer
- Latency in milliseconds of FCoE, iSCSI, NFS, and FC protocols as reported by Iometer
- Average vSphere 4.1 server CPU utilization as reported by the vSphere client

SINGLE VM GENERATING HIGH LEVELS OF CONCURRENT I/O USING A NONSHARED DATASTORE

The workload used for the single VM test was a mixture of random reads and writes using an 8K request size to represent an OLTP type of database application workload in a production environment running high levels of concurrent I/O from a single VM. The specifics of this load are as follows:

- 8K request size, 60% read, 40% write, 100% random

When conducting the performance tests involving a single VM using either LSI Logic or VMware PVSCSI controller drivers for any of the protocols, we used the identical storage and vSphere datastore configurations defined in section 6.3 through section 6.4. For these tests, we removed the two 10GB virtual disks used for the initial testing from one of the VMs on vSphere server 1 and added two 200GB virtual disks, one on each of the FAS6210 storage controllers. This was done to better reflect the larger file sizes that would be associated with a production database and make the data size large enough that the data could not be cached in the storage controller's memory during the test run.

Depending on the SCSI controller driver being tested, we selected either LSI Logic or VMware PVSCSI controller drivers for these two 200GB virtual disks in the vSphere client. We then assigned drive letters D:\ and E:\ to these newly created virtual disks using the Windows disk manager from within the guest OS. Additionally, we configured this single VM with four virtual CPUs instead of the one virtual CPU that was used for testing utilizing multiple VMs. We selected four virtual CPUs for these tests so that the VM CPU resources would not be a bottleneck during the test.

After the two 200GB virtual disks were created, and drive letters D:\ and E:\ assigned, we initialized the virtual disks in this single VM with Iometer data files for use during the performance tests. The VM was booted, and Iometer was presented with its corresponding set of Iometer disk targets (D:\ and E:\).

The following steps were executed to initialize Iometer target disks:

1. Powered on the single VM1 on vSphere server 1.
2. Ran the Iometer Dynamo application in the single VM.
3. Launched the Iometer controller application on the server designated as the Iometer controller server.
4. In the Iometer controller application, selected both drive letters corresponding to the Iometer target disks.
5. Created an Iometer access specification to generate a 100% random-read load using 4K request size.
6. Executed the specification on the single VM for a total of 5 minutes.

Again, the results of these random-read test runs were discarded because the test served only to create a 200GB Iometer data file in the virtual disks associated with the VM.

After initializing the virtual disks seen by the Iometer controller, we powered off the single VM and created a Snapshot copy of the aggregate defined in Tables 5 and 6. This Snapshot copy represented the initial state of the system before we performed any tests that modified the data files. At the end of every test, we powered off the single VM and used this Snapshot copy to restore the storage to the freshly initialized state so that all tests would be started from exactly the same point in time with the Iometer data files in their initialized state.

For testing with FCoE, iSCSI, and FC, the same set of LUNs was used to hold all VM OS virtual disks, and another set of LUNs was used to hold the VM's VMDK files containing the Iometer data files. Therefore, in order to change the block protocol under test, we unmapped/remapped the relevant LUNs holding the VM boot OS disks and VMDK files containing the Iometer data files to an initiator group that corresponded to the protocol being tested. We then rescanned for datastores in the vSphere 4.1 client so that the datastores were seen by the vSphere 4.1 hosts using the corresponding protocol being tested.

For the single VM tests, the Iometer access specification was configured so that the single Dynamo instance running on the VM generated a constant total of 128 outstanding I/Os. Because the VM consisted of four virtual CPUs, there were four workers configured for the single Dynamo instance. Each worker was configured to drive 16 outstanding I/Os to drives D:\ and E:\. This resulted in 64 outstanding I/Os being driven to each drive for a total of 128 outstanding I/Os spread evenly across the 2 drives. Additionally, the total size of the Iometer working data-set size was 400GB. This working set was large enough so that the data was not served from the cache on the FAS6210A, resulting in modest to high levels of disk activity on the FAS6210A.

For all test cases, the Iometer tests were run for a total of 15 minutes, with a 10-minute ramp-up time to allow performance to reach a steady state before Iometer began to record results.

7 CONCLUSION

We believe these tests demonstrate that the combination of vSphere 4.1 and the NetApp unified storage platform provides enterprise-class performance in a variety of typical production scenarios with any of the protocols supported by VMware and NetApp. The large number of protocols supported clearly provides the ultimate in flexibility for our customers to move forward with emerging data center standards like Data Center Ethernet while maintaining the viability of their existing vSphere environments.

Additionally, we found that using the PVSCSI driver in our high-I/O environment allowed us to generate performance comparable to that of LSI Logic while using significantly fewer VM CPU resources.

Finally, we found that the differences in vSphere host CPU resources consumed by the different protocols during the shared datastore testing were generally in the range of 3% or less and deemed statistically irrelevant. Therefore, the performance engineering teams at VMware and NetApp agreed to omit comparative charts from this report.

8 REFERENCES

- TR-3747: "Best Practices for File System Alignment in Virtual Environments"
  http://media.netapp.com/documents/tr-3747.pdf
- TR-3749: "NetApp and VMware vSphere Storage Best Practices"
  http://media.netapp.com/documents/tr-3749.pdf
- WP-7123: "NetApp Unified Connect: Taking Convergence to the Next Level"
  http://media.netapp.com/documents/wp-7123.pdf

9 ACKNOWLEDGEMENTS

Special thanks to the following for their contributions:

- Duaine Fiorrito, Solutions Technologies Lab Manager, NetApp
- Keith Griffin, EDCA Lab Support, NetApp
- Christopher Lemmons, Senior Manager, Workload Engineering, NetApp
- Vaughn Stewart, Director and Evangelist for Virtualization and Cloud Computing, NetApp

10 FEEDBACK

Send an e-mail to xdl-vgibutmevmtr@netapp.com with questions or comments concerning this document.