



Technical Report

The Benefits of Converged Networking with NetApp FAS6280 and Intel Xeon Servers

Benjamin Krieger, NetApp
January 2011 | TR-3894

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	TARGET AUDIENCE.....	4
2	MEETING THE DENSITY REQUIREMENTS OF AN INCREASINGLY VIRTUALIZED COMPUTING ENVIRONMENT	4
2.1	RECOMMENDED SUPPORT DOCUMENTS.....	5
3	SOLUTION ARCHITECTURE.....	6
3.1	HARDWARE REQUIREMENTS.....	6
3.2	SOFTWARE REQUIREMENTS	6
3.3	SOLUTION ARCHITECTURE DETAILS	7
4	STORAGE PROVISIONING	7
5	ISCSI	8
5.1	BEST PRACTICES	8
5.2	SWITCH SETUP	9
5.3	STORAGE SYSTEM SETUP	9
5.4	SERVER SETUP.....	10
6	FCOE.....	15
6.1	SWITCH SETUP	15
7	CONFIGURE ETHERNET INTERFACES USED FOR FCOE INITIATOR.....	17
8	CONFIGURE ETHERNET INTERFACES USED FOR FCOE TARGET	17
9	CREATE VFC FOR INITIATOR	18
10	CREATE VFC FOR TARGET	18
10.1	ADD VFC TO VSAN	18
10.2	CREATE ZONESETS AND ZONES.....	19
10.3	CREATE A CLASS OF SERVICE.....	20
10.4	STORAGE SYSTEM SETUP	21
10.5	SERVER SETUP.....	22
11	RESULTS AND VALIDATION.....	23
11.1	IOMETER TESTS	23
11.2	UNIFIED CONNECT (FCOE AND CIFS) TESTS	31
11.3	SQLIOSIM TESTS	35
11.4	SPACE RECLAIMER.....	37
11.5	SNAPMANAGER FOR MICROSOFT SQL SERVER.....	38
12	SUMMARY	41
13	REFERENCES	42
14	ACKNOWLEDGMENTS.....	42

LIST OF TABLES

Table 1) Part 1 of test results.....	36
Table 2) Part 2 of test results.....	36

LIST OF FIGURES

Figure 1) Environment overview.....	5
Figure 2) Data ONTAP logical view.....	7
Figure 3) Fabric view.....	22
Figure 4) iSCSI: 100% read, 100% sequential.....	24
Figure 5) iSCSI: 100% write, 100% sequential.....	25
Figure 6) iSCSI: 100% read, 100% random.....	25
Figure 7) iSCSI: 100% write, 100% random.....	26
Figure 8) iSCSI: 50% read, 100% sequential.....	26
Figure 9) iSCSI: 50% read, 100% random.....	27
Figure 10) iSCSI: 50% read, 50% random.....	27
Figure 11) FCoE: 100% read, 100% sequential.....	28
Figure 12) FCoE: 100% write, 100% sequential.....	28
Figure 13) FCoE: 100% read, 100% random.....	29
Figure 14) FCoE: 100% write, 100% random.....	29
Figure 15) FCoE: 50% read, 100% sequential.....	30
Figure 16) FCoE: 50% read, 100% random.....	30
Figure 17) FCoE: 50% read, 50% random.....	31
Figure 18) FCoE and CIFS: 100% read, 100% sequential.....	32
Figure 19) FCoE and CIFS: 100% write, 100% sequential.....	32
Figure 20) FCoE and CIFS: 100% read, 100% random.....	33
Figure 21) FCoE and CIFS: 100% write, 100% random.....	33
Figure 22) FCoE and CIFS: 50% read, 100% sequential.....	34
Figure 23) FCoE and CIFS: 50% read, 100% random.....	34
Figure 24) FCoE and CIFS: 50% read, 50% random.....	35

1 INTRODUCTION

Consolidating networks in the data center into a unified, integrated infrastructure, referred to as network convergence, reduces both the overall capital expenditure required for network deployment and the operational expenditure for maintaining the infrastructure. Customers can obtain the benefits of network convergence with newly released products from both NetApp and Intel—the NetApp® FAS6280 and the Intel® Xeon™.

In addition to the increased power offered by the Intel Xeon-based servers and NetApp storage systems, converged networking can reduce the footprint, price, and complexity commonly associated with deploying high-speed centralized storage throughout a computing environment.

Furthermore, converged networking gives storage and system architects their choice of protocols for accessing storage, and when deployed on a 10-Gigabit Ethernet (GbE) network, both Fibre Channel over Ethernet (FCoE) and iSCSI provide uncompromisingly low latency and high throughput. With software-driven Ethernet network interface cards (NICs) increasingly available in servers, administrators can react quickly and flexibly to changing requirements without the need for additional hardware.

This paper provides architectural and deployment details for each part of a demonstration environment that uses the recently available unified networking (including Unified Connect) and unified storage technologies. Best practices, setup instructions, and efficiency features are also discussed. This paper does not provide performance benchmarks; rather, the functional testing discussed in this paper demonstrates the general equivalency and effectiveness of the available protocol options.

1.1 TARGET AUDIENCE

This paper is recommended for system and storage architects who are interested in:

- Upgrading to or deploying a highly dense or cloud computing environment
- Increasing the density of an existing environment
- Understanding converged networking technologies

2 MEETING THE DENSITY REQUIREMENTS OF AN INCREASINGLY VIRTUALIZED COMPUTING ENVIRONMENT

The movement toward increased density in virtualized and cloud computing environments has been steadily accelerating since the widespread adoption of hypervisor-enabled systems. Newly released products from both NetApp and Intel will accelerate this trend. The NetApp FAS6280 storage system can handle large amounts of simultaneous requests from hypervisor-enabled servers, and the Intel Xeon 5600 series processors provide an increased amount of processing power on a per-server basis.

The NetApp FAS6280 storage system running Data ONTAP® 8.0.1 can address a maximum of 1,440 disks and 2.8 petabytes of storage. With 12 cores, 96GB of onboard memory, and Flash Cache capability, it can service a large number of simultaneous requests from hypervisor-enabled servers.

Likewise, Intel servers using Xeon 5600 series processors contain four or more cores per socket. With server-class motherboards typically containing two to four sockets and each core capable of simultaneously executing two threads, the amount of processing power available on a per-server basis is growing rapidly.

To keep pace, networking must also increase its density by increasing the bandwidth available per port and by consolidating the communication protocols into one adapter.

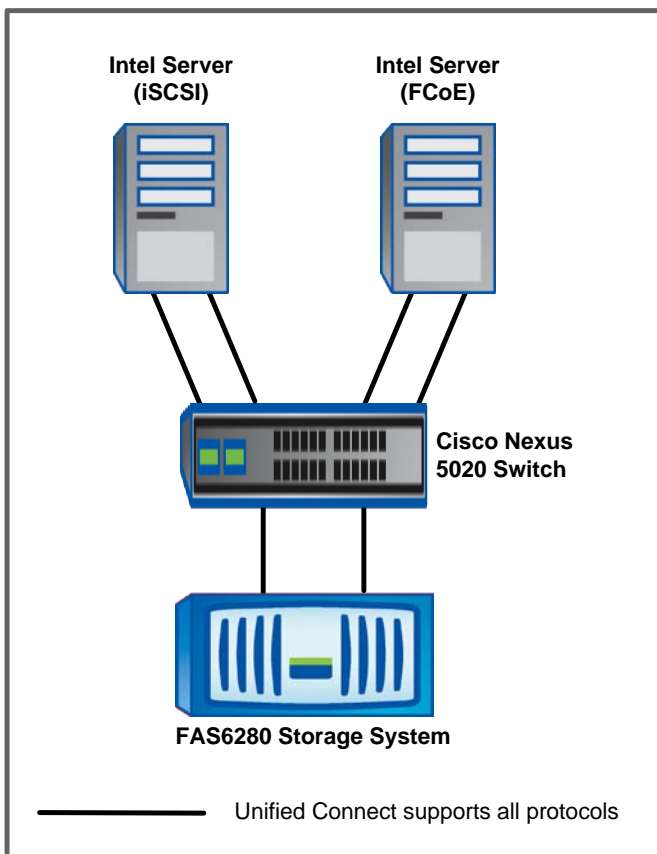
NetApp storage servers that use X1139/X1140 unified target adapters have more available on-board ports for access to disk shelves. In addition, more expansion card slots are left open for use by other performance-enhancing technologies such as Flash Cache.

Likewise, on the server side, Intel X520 Ethernet adapters are capable of both standard TCP/IP traffic and FCoE traffic simultaneously. The X520 10GbE adapter implements the FCoE protocol in software by using the OpenFCoE software stack; therefore, a lower up-front investment in connectivity hardware is required even for highly dense environments.

Finally, the Cisco® Nexus line of switches reduces the footprint of both cabling and networking equipment by implementing both a traditional Ethernet switch and a Fibre Channel (FC) fabric with Data Center Bridging (DCB) capability in the same piece of hardware. A prerequisite for FC connectivity is a lossless medium, and DCB achieves losslessness by permitting FCoE to be prioritized over other traffic or by assigning a given percentage of available bandwidth in a class of service (CoS).

This paper examines each part of a demonstration environment. An overview of the environment is shown in Figure 1. As previously mentioned, best practices, setup instructions, and efficiency features are discussed. The functional testing, discussed in this paper, demonstrates the general equivalency and effectiveness of the available protocol options.

Figure 1) Environment overview.



2.1 RECOMMENDED SUPPORT DOCUMENTS

Refer to these documents for more information on this solution:

- [Intel Cloud Builders Guide: Cloud Design and Deployment on Intel Platforms](#)
- [TR-3441: Windows Multipathing Options with Data ONTAP—FCP and iSCSI](#)
- [TR-3800: Fibre Channel over Ethernet \(FCoE\) End-to-End Deployment Guide](#)

3 SOLUTION ARCHITECTURE

The solution discussed in this paper consists of dual-path 10GbE connections from the Intel-based servers to the back-end storage system through the use of a Cisco Nexus 5020 switch. On the server demonstrating iSCSI connectivity, the native Microsoft® iSCSI initiator software is used. On the server demonstrating FCoE connectivity, the OpenFCoE driver stack is used. The multipath configuration is facilitated by the NetApp DSM on the FCoE-based server and by the native MPIO drivers on the iSCSI-based server.

3.1 HARDWARE REQUIREMENTS

This solution uses the following hardware:

- Intel white box dual-processor server
 - Two Intel Xeon 5680 processors
 - Six cores with a total of 12 threads each
 - 3.33GHz clock speed
 - Intel X520 series 10Gb/s Ethernet adapter
 - 24GB memory
 - 2x146GB HDD RAID 0
- NetApp FAS6280 high-end storage system
 - X1139 unified target adapter
 - 2TB Flash Cache
 - Three DS14mk4 shelves with 14 FC 450GB drives each
 - Two DS4243 shelves with 24 SAS 450GB drives each
 - One DS4243 shelf with 24 SATA 1TB drives
- Cisco Nexus 5020 unified switch

3.2 SOFTWARE REQUIREMENTS

This solution requires the following software:

- Server operating system: Microsoft Windows® 2008R2 Enterprise Edition
- Storage platform operating system: Data ONTAP 8.0.1
- Switch firmware revision: Cisco NX-OS 4.2
- Benchmarking and test utilities:
 - IOmeter version 2006.07.27
 - Microsoft SQLIOSim
- Support and administration utilities:
 - NetApp SnapDrive® 6.3
 - NetApp SnapManager® for Microsoft SQL Server® 5.1
 - NetApp DSM 3.3.1
- Application software: Microsoft SQL Server 2008 SP1

3.3 SOLUTION ARCHITECTURE DETAILS

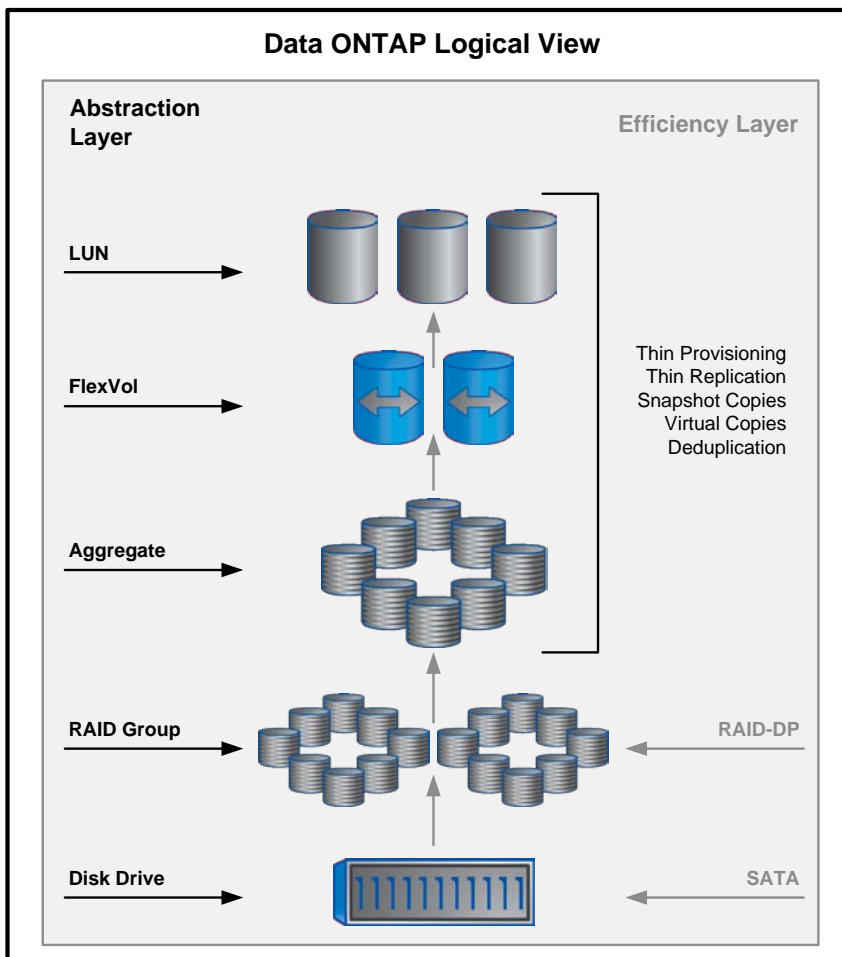
The steps for the pretest setup and provisioning of the storage system, servers, and network are as follows:

1. The aggregates and the NetApp FlexVol[®] volumes are created on the storage system.
2. The storage system and the servers are linked together using a Cisco Nexus switch capable of supporting Ethernet and Fibre Channel protocols at 10GbE speeds.
3. NetApp SnapDrive software is used to create a logical unit number (LUN) on the previously provisioned FlexVol volumes, perform LUN masking, and create an NTFS file system.

4 STORAGE PROVISIONING

Figure 2 shows a logical view of how storage is provisioned from a disk drive to a LUN.

Figure 2) Data ONTAP logical view.



To accommodate both the line-rate and application functional testing, we provisioned the storage using the following steps.

1. Create an aggregate to contain the test FlexVol volumes. Use a RAID-DP[®] layout for the aggregate RAID groups. This method reduces the risk of data loss, prevents the parity from becoming a

performance bottleneck, and makes sure that a RAID group can survive the loss of more than one disk.

```
aggr create aggr1 -t raid_dp -T SAS 23
```

2. Create flexible volumes to contain the test LUNs. To mimic a standard SQL Server environment, we provisioned one LUN for data (2TB) and one LUN for logs (200GB), and each LUN is placed in a separate FlexVol volume. The space reserve is turned off in the FlexVol volumes; therefore, the free space of the containing aggregate is shared, and unallocated space is not wasted.

```
vol create sql_fcoe_log aggr1 -s none 200G
vol create sql_fcoe_data aggr1 -s none 2T
vol create sql_iscsi_log aggr1 -s none 200G
vol create sql_iscsi_data aggr1 -s none 2T
```

3. Turn off the Snapshot[®] schedule and set the snap reserve percentage to zero. This step takes advantage of the Data ONTAP thin provisioning capabilities and allows space management at the aggregate rather than at the volume level. Furthermore, the fractional reserve or overwrite protection percentage is set to zero.

Note: The Snapshot schedule can also be turned off when creating LUNs in the SnapDrive Create Disk wizard.

```
vol option set sql_fcoe_log fractional_reserve 0
vol option set sql_fcoe_data fractional_reserve 0
vol option set sql_iscsi_log fractional_reserve 0
vol option set sql_iscsi_data fractional_reserve 0
snap sched sql_fcoe_log 0 0 0
snap sched sql_fcoe_data 0 0 0
snap sched sql_iscsi_log 0 0 0
snap sched sql_iscsi_data 0 0 0
snap reserve sql_fcoe_log 0
snap reserve sql_fcoe_data 0
snap reserve sql_iscsi_log 0
snap reserve sql_iscsi_data 0
```

5 ISCSI

5.1 BEST PRACTICES

The configuration followed iSCSI connectivity best practices:

- Enable Receive Side Scaling on the host X520 Ethernet adapter.
 - This feature spreads the CPU requirements for the iSCSI software initiator across the large number of processor cores available in systems that use Intel Xeon processors, thus increasing efficiency and scalability. This feature is enabled by default in Windows Server 2008 R2.
- Enable Large Segment Offload on the host X520 Ethernet adapter.
 - This feature moves the work of splitting large data requests into segments small enough to be transmitted by the IP protocol to the Ethernet adapter. This feature is enabled by default in Windows Server 2008 R2.
- Enable jumbo frames of 9,000 bytes on the host, switch, and storage.
 - This reduces the amount of network overhead required for TCP/IP headers and checksums because it decreases the ratio of metadata to data in each packet sent or received over the standard MTU size of 1,500 bytes.

- Establish multiple TCP streams between the host and storage, both by using a multiconnection session (MCS) and a standard Microsoft Multipath I/O (MPIO) implementation shipped with Windows Server 2008 R2.
 - MCS allows multiple TCP streams with the same endpoints to carry iSCSI traffic. This feature improves performance because it adds parallelism.
 - MPIO demonstrates a similar performance boost, with the added benefit of a redundant path from the server to the storage system.

5.2 SWITCH SETUP

To set up the switch, we only had to configure the interfaces for the Ethernet traffic and enable a service policy with jumbo frames.

1. Create a network-qos-type policy map named “jumbo.”

```
switch(config)# policy-map type network-qos jumbo
```

2. Set the class type to class-default (this class is created on startup by NX-OS and cannot be removed).

```
switch(config-pmap-nq)# class type network-qos class-default
```

3. Set the frame size to 9216, which is large enough to accommodate jumbo frames.

```
switch(config-pmap-nq)# mtu 9216
```

4. Configure and enable QoS policies.

```
switch(config-pmap-nq)# system qos
```

5. Set the network-qos service policy named “jumbo.”

```
switch(config-sys-qos)# service-policy type network-qos jumbo
```

5.3 STORAGE SYSTEM SETUP

To use the storage system as a target for iSCSI traffic, we set up the e3a and e5a interfaces belonging to the X1139 unified target adapters as follows:

1. Configure both 10GbE interfaces with Ethernet IP addresses, along with the appropriate netmask and broadcast addresses.

```
filer> ifconfig e3a 10.17.253.111 netmask 255.255.255.0 broadcast
10.17.253.255
filer> ifconfig e5a 10.17.253.113 netmask 255.255.255.0 broadcast
10.17.253.255
```

2. Set the MTU size for both of the interfaces to 9000 to enable jumbo frame capabilities and to increase potential throughput.

```
filer> ifconfig e3a mtusize 9000
filer> ifconfig e5a mtusize 9000
```

3. Disable all interfaces not meant for use as an iSCSI target, and verify the results.

```
filer> iscsi interface disable e0a
filer> iscsi interface disable e0b
...
filer> iscsi interface disable e5b
filer> iscsi interface show
```

The results are as follows.

```
Interface e0M disabled
Interface e0a disabled
Interface e0b disabled
Interface e0c disabled
Interface e0d disabled
Interface e0e disabled
Interface e0f disabled
Interface e3a enabled
Interface e3b disabled
Interface e5a enabled
Interface e5b disabled
```

5.4 SERVER SETUP

Preparing a server for iSCSI access to back-end data on a storage system requires three steps.

1. Configure the network, including the recommended NIC settings.
2. Establish the iSCSI initiator-to-target relationship (by using the Microsoft iSCSI initiator).
3. Create and map the LUN to the server's initiator by the storage system (by using NetApp SnapDrive).

CONFIGURE THE ETHERNET NIC

The previously described best practices can be confirmed by examining the NIC device properties. Complete the following steps to access the properties pane:

1. Launch the Server Manager utility.
2. Navigate to Diagnostics, Device Manager, Network Adapters.
3. Right-click the interface in question.
4. Select Properties.
5. Set the parameters on each interface individually, even if they are part of the same physical expansion card.
6. Check the setting for Receive Side Scaling on the Advanced tab and make sure it is set to Enabled.

CONFIGURE THE ISCSI INITIATOR-TARGET

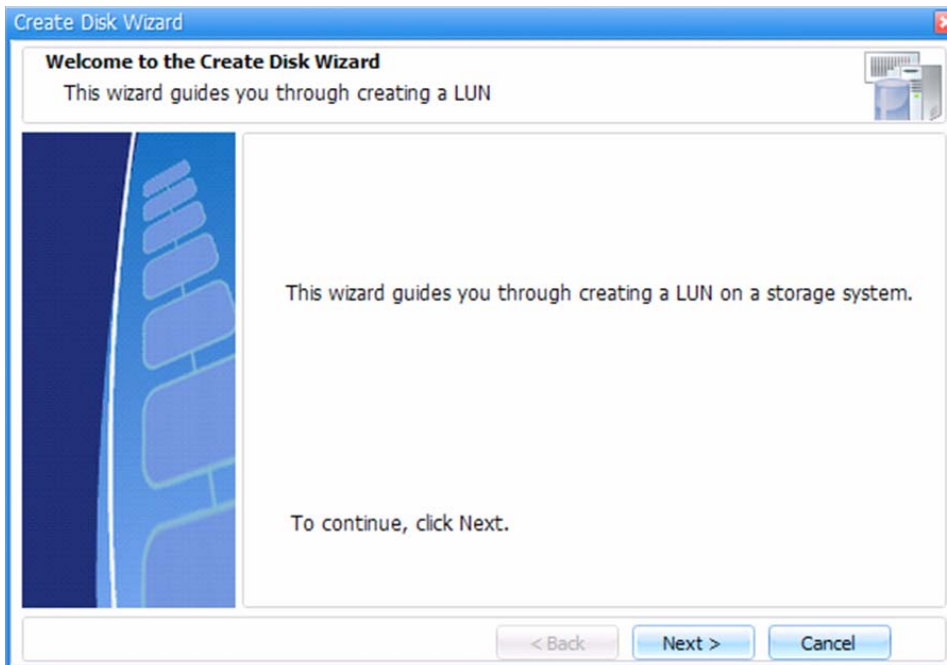
Complete the following steps to configure the iSCSI Initiator-Target:

1. Open the iSCSI Initiator tool from Start, Administrative Tools, iSCSI Initiator.
2. Navigate to the Discovery tab, and then click Discover Portal.
3. Enter the IP address of the first iSCSI interface on the storage system, and then click OK.
4. Repeat step 3 for the second iSCSI interface on the storage system. Both IP addresses should show up in the Discovery tab.
5. Navigate to the Targets tab and click Connect.
6. Add this connection to the list of Favorite Targets in the Connect to Target dialog box, enable multipath, and then click Advanced Settings.
7. Select the IP address in the Connect Using section of the server's first interface under the Initiator IP. Select the IP address of the storage system's first interface (e3a in our example) under Target portal IP. Click OK.
8. Click OK to connect to the target.

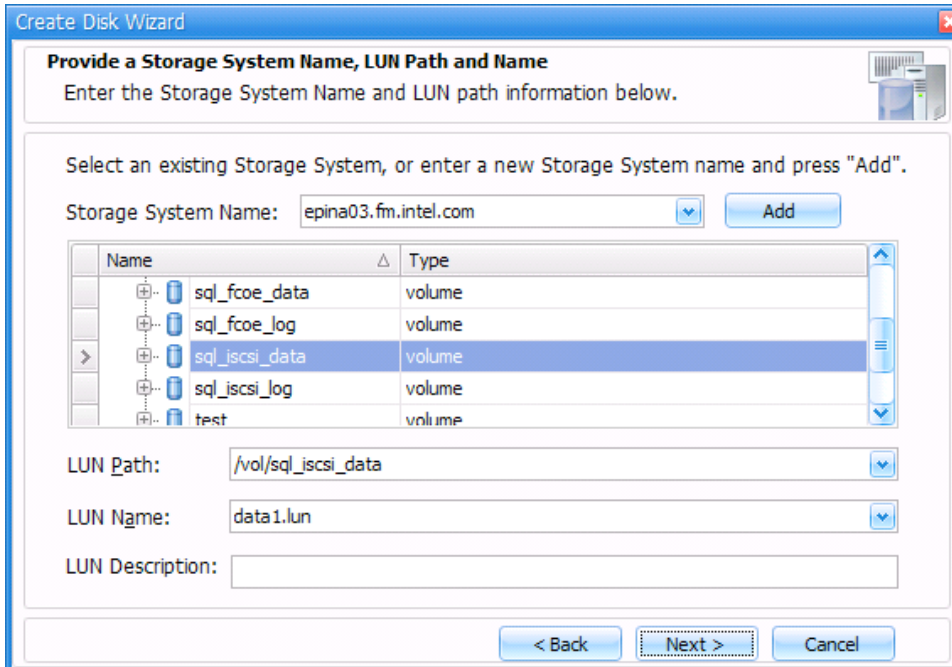
9. Click Properties after highlighting the now-connected target in the Targets tab, and then click Add Session.
10. Repeat steps 6–8, this time using the server's second interface IP as the Initiator IP and the storage system's second interface (e5a in our example).
11. Verify that more than one session identifier is listed in the Properties pane, and click MCS... to configure multiple connections per session.

CREATE AND CONFIGURE LUN

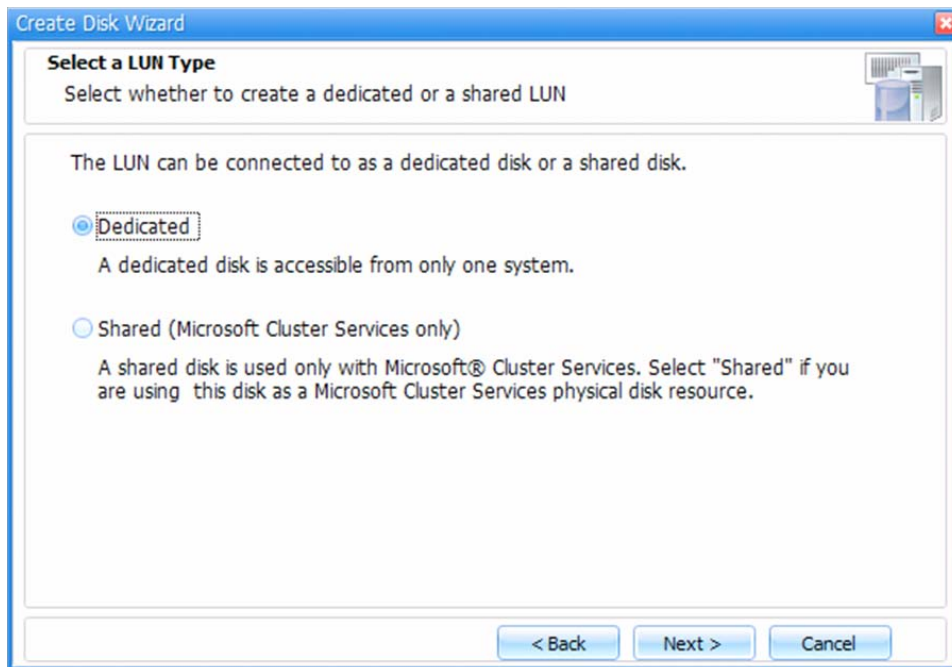
1. Start the SnapDrive Create Disk wizard.
2. Click Next.



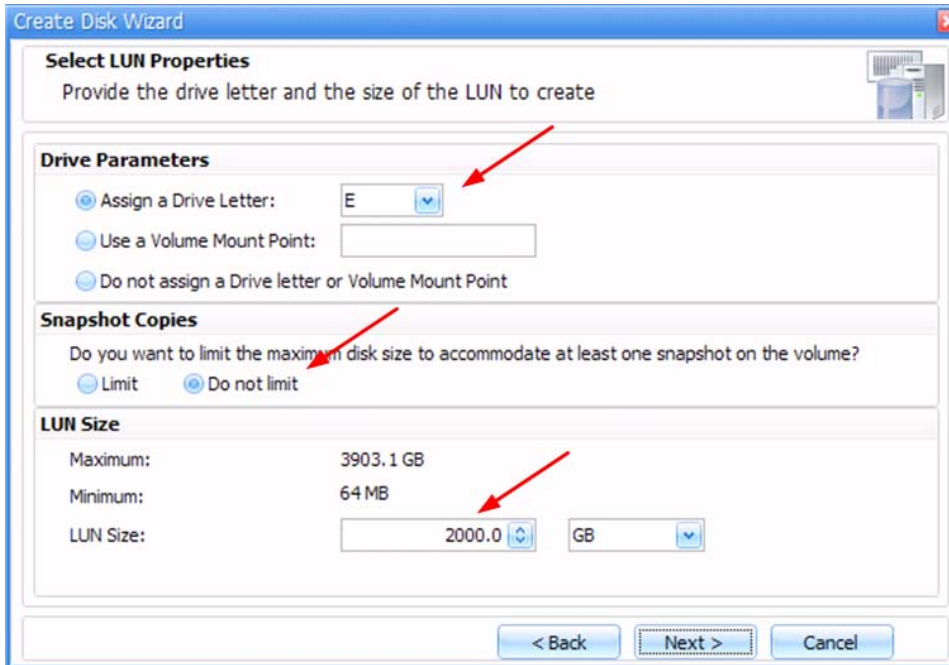
3. Enter the Storage System name, LUN Path, and LUN Name. An optional LUN description can also be added.
4. Click Next.



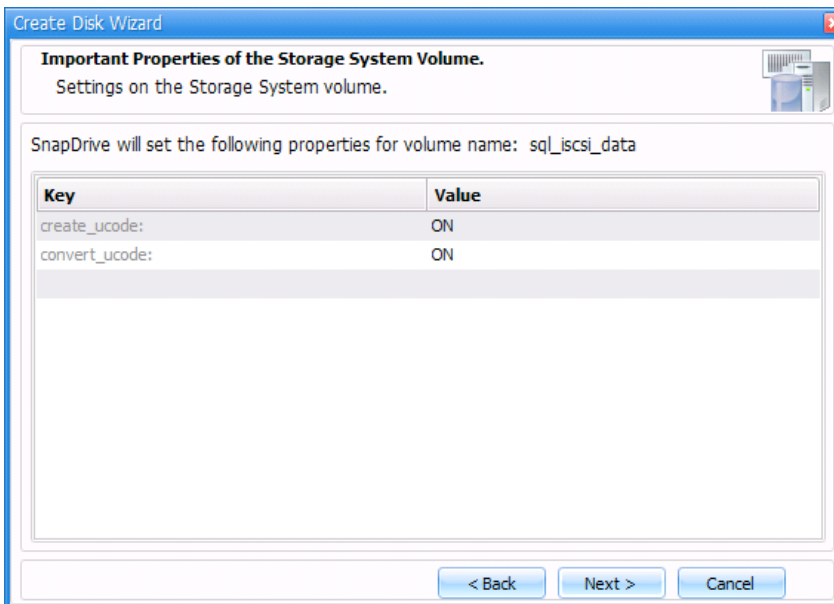
5. Select Dedicated as the LUN type.
6. Click Next.



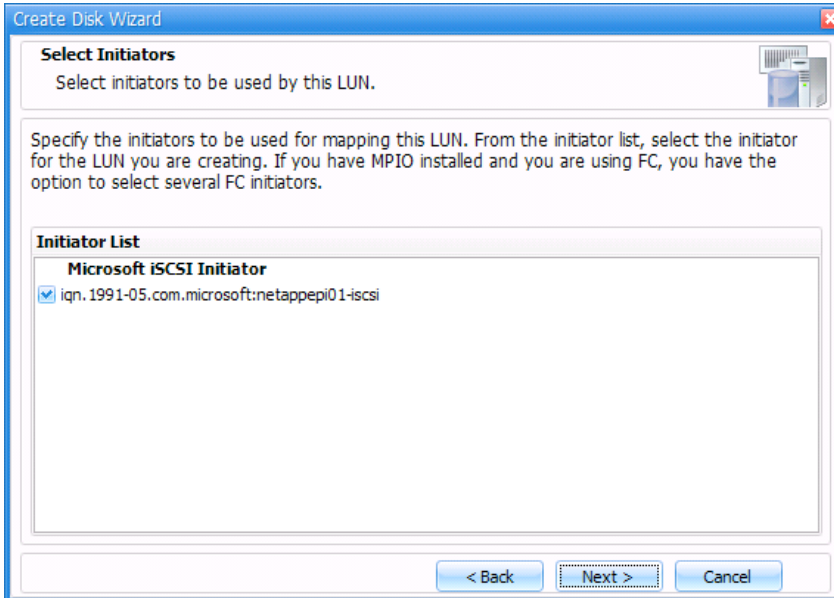
7. Specify the LUN properties including which Drive Letter to assign.
8. Disable the Snapshot copies limitation by selecting Do not limit.
9. Set the LUN Size to the size laid out in the Storage Provisioning section.
10. Click Next.



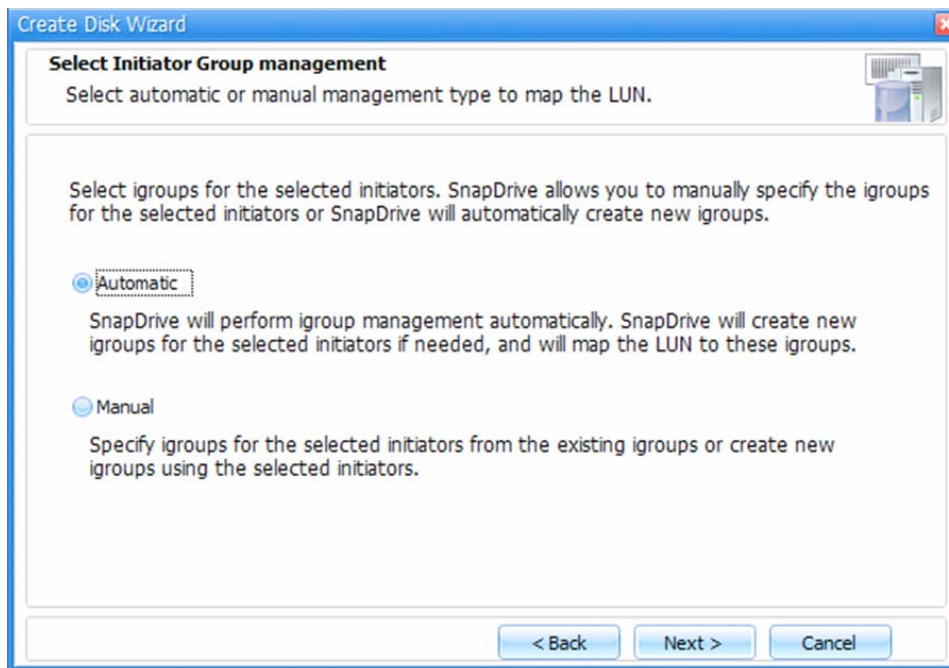
11. Set the Storage System Volume properties for the LUN.
12. Click Next.



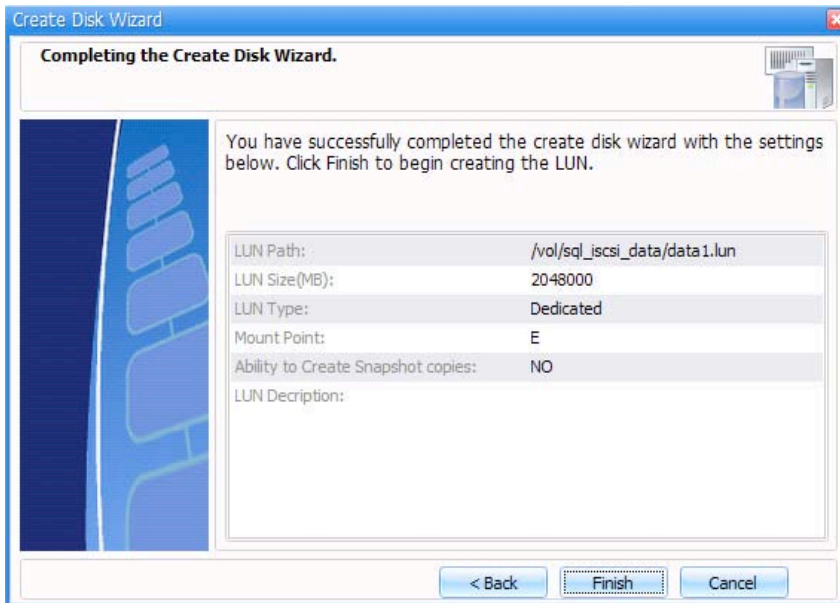
13. Pick the iSCSI initiators from the Initiator List.
14. Click Next.



15. Select Automatic for the igroup management. This inserts the previously selected initiators into the appropriate igroup on the storage system to make the newly created LUN visible.
16. Click Next.



17. Review the LUN parameters, and then click Finish.



6 FCOE

6.1 SWITCH SETUP

After the server initiators and the storage system targets are in the same zone in the FC fabric, only a few more steps are required to enable FCoE connectivity. Set up the switch between the server and the storage system so the initiator and target become visible to each other.

Ports 3b and 5b on the storage system are used for FCoE connectivity.

CREATE A VSAN

1. Make sure the FCoE feature is licensed before starting the configuration.
2. Create a VSAN to act as a logical container for the target and initiator ports.

```
switch# show vsan
vsan 1 information
    name:VSAN0001  state:active
    interoperability mode:default
    loadbalancing:src-id/dst-id/oxid
    operational state:down

vsan 4079:evfp_isolated_vsan

vsan 4094:isolated_vsan

switch# configure terminal

switch(config)# vsan database

switch(config-vsan-db)# vsan 01000
switch(config-vsan-db)# end
```

```

switch# show vsan
vsan 1 information
    name:VSAN0001 state:active
    interoperability mode:default
    loadbalancing:src-id/dst-id/oxid
    operational state:down

vsan 1000 information
    name:VSAN1000 state:active
    interoperability mode:default
    loadbalancing:src-id/dst-id/oxid
    operational state:down

vsan 4079:evfp_isolated_vsan

vsan 4094:isolated_vsan

```

CREATE A VLAN AND MAP IT TO THE NEW VSAN

Create a VLAN and map it to the new VSAN.

```

switch# show vlan

VLAN Name                Status      Ports
-----
--
1      default              active     Eth1/1, Eth1/2, Eth1/3,
Eth1/4                                Eth1/5, Eth1/6, Eth1/7,
Eth1/8                                Eth1/9, Eth1/10, Eth1/11
                                         Eth1/12, Eth1/13, Eth1/14
                                         Eth1/15, Eth1/16, Eth1/17
                                         Eth1/18, Eth1/19, Eth1/20

Remote SPAN VLANs
-----
--

Primary  Secondary  Type      Ports
-----
---

switch# configure terminal
switch(config)# vlan 1000
switch(config-vlan)# fcoe vsan 1000
switch(config-vlan)# end

switch# show vlan

VLAN Name                Status      Ports
-----
--
1      default              active     Eth1/1, Eth1/2, Eth1/3,
Eth1/4

```



```

Eth1/8
Eth1/5, Eth1/6, Eth1/7,
Eth1/9, Eth1/10, Eth1/11
Eth1/12, Eth1/13, Eth1/14
Eth1/15, Eth1/16, Eth1/17
Eth1/18, Eth1/19, Eth1/20

1000 VLAN1000 active
VLAN Name Status Ports
-----
--

Remote SPAN VLANs
-----
--

Primary Secondary Type Ports
-----
---
switch# show vlan fcoe
VLAN VSAN Status
-----
1000 1000 Operational

```

7 CONFIGURE ETHERNET INTERFACES USED FOR FCOE INITIATOR

Set the spanning-tree protocol (STP) type and assign a VLAN for the Ethernet ports used for the initiator.

```

switch# configure terminal

switch(config)# interface Ethernet 1/1

switch(config-if)# switchport mode trunk

switch(config-if)# switchport trunk allowed vlan 1,1000

switch(config-if)# spanning-tree port type edge trunk

Warning: Edge port type (portfast) should only be enabled on ports connected
to a single host. Connecting hubs, concentrators, switches, bridges, etc...
to this interface when edge port type (portfast) is enabled, can cause
temporary bridging loops.
Use with CAUTION

switch(config-if)# end

```

8 CONFIGURE ETHERNET INTERFACES USED FOR FCOE TARGET

Set the spanning-tree port type and assign a VLAN for the Ethernet ports used for the target.

```

switch# configure terminal

switch(config)# interface Ethernet 1/16

switch(config-if)# switchport mode trunk

```

```
switch(config-if)# switchport trunk allowed vlan 1,1000
```

```
switch(config-if)# spanning-tree port type edge trunk
```

Warning: Edge port type (portfast) should only be enabled on ports connected to a single host. Connecting hubs, concentrators, switches, bridges, etc... to this interface when edge port type (portfast) is enabled, can cause temporary bridging loops.

Use with CAUTION

```
switch(config-if)# end
```

9 CREATE VFC FOR INITIATOR

A Virtual Fibre Channel (VFC) interface can be bound to a specific Ethernet port or to a worldwide port name (WWPN). In this example, we are binding to an Ethernet port. The VFC number does not have to match the physical port number, but this improves manageability. Create a VFC for the initiator.

```
switch# configure terminal
```

```
switch(config)# interface vfc 1
```

```
switch(config-if)# bind interface ethernet 1/1
```

```
switch(config-if)# end
```

10 CREATE VFC FOR TARGET

Create a VFC for the target.

```
switch# configure terminal
```

```
switch(config)# interface vfc 16
```

```
switch(config-if)# bind interface ethernet 1/16
```

```
switch(config-if)# end
```

10.1 ADD VFC TO VSAN

Add any FC interfaces that communicate with these FCoE interfaces to the VSAN as well.

```
switch# configure terminal
```

```
switch(config)# vsan database
```

```
switch(config-vsan-db)# vsan 1000 interface vfc 1
```

```
switch(config-vsan-db)# vsan 1000 interface vfc 16
```

```
switch(config-vsan-db)# end
```

```
switch# show vsan membership
```

```

vsan 1 interfaces:
    fc2/1          fc2/2          fc2/3          fc2/4
    fc2/5          fc2/6          fc2/7          fc2/8

vsan 1000 interfaces:
    vfc1           vfc16

vsan 4079(evfp_isolated_vsan) interfaces:

vsan 4094(isolated_vsan) interfaces:

```

10.2 CREATE ZONESETS AND ZONES

Create zonesets and zones.

```

switch# configure terminal

switch(config)# show flogi database vsan 1000
-----
---
INTERFACE          VSAN    FCID          PORT NAME          NODE NAME
-----
---
vfc1                1000    0x470000     21:00:00:c0:dd:12:04:cf
20:00:00:c0:dd:12:04:cf
vfc16               1000    0x470001     50:0a:09:85:89:8b:4f:5c
50:0a:09:80:89:8b:4f:5c

Total number of flogi = 2.

switch(config)# show fcns database vsan 1000

VSAN 1000:
-----
FCID          TYPE  PWWN          (VENDOR)          FC4-TYPE:FEATURE
-----
0x470000     N     21:00:00:c0:dd:12:04:cf (Qlogic)          scsi-fcp:init
0x470001     N     50:0a:09:85:89:8b:4f:5c (NetApp)          scsi-fcp:target

Total number of entries = 2

switch(config)# zone name fcoezonel vsan 1000

switch(config-zone)# member pwwn 21:00:00:c0:dd:12:04:cf

switch(config-zone)# member pwwn 50:0a:09:85:89:8b:4f:5c
switch(config-zone)# zoneset name fcoe vsan 1000

switch(config-zoneset)# zone name fcoezonel

switch(config-zoneset-zone)# zoneset activate name fcoe vsan 1000

Zoneset activation initiated. check zone status

```

```

switch(config)# end

switch# show zoneset active vsan 1000

zoneset name fcoe vsan 1000
  zone name fcoezone1 vsan 1000
  * fcid 0x470000 [pwwn 10:00:00:00:c9:9c:6c:93]
  * fcid 0x470001 [pwwn 50:0a:09:85:89:8b:4f:5c]

```

10.3 CREATE A CLASS OF SERVICE

By default, FCoE is allocated 50% of available bandwidth using DCB to provide the lossless performance required by the FC protocol. An additional, optional step is to create and then assign a CoS that allocates a different amount of bandwidth to FCoE. In this example, the FCoE is assigned a policy that allocates 80% bandwidth.

```

switch# show policy-map
Type qos policy-maps
=====

policy-map type qos default-in-policy
  class type qos class-fcoe
    set qos-group 1
  class type qos class-default
    set qos-group 0

Type queuing policy-maps
=====

policy-map type queuing default-in-policy
  class type queuing class-fcoe
    bandwidth percent 50
  class type queuing class-default
    bandwidth percent 50
policy-map type queuing default-out-policy
  class type queuing class-fcoe
    bandwidth percent 50
  class type queuing class-default
    bandwidth percent 50

Type network-qos policy-maps
=====

policy-map type network-qos default-nq-policy
  class type network-qos class-fcoe
    pause no-drop
    mtu 2158
  class type network-qos class-default
    mtu 1500

switch# configure terminal

switch(config)# policy-map type queuing fcoe-80

switch(config-pmap-que)# class type queuing class-fcoe

```

```

switch(config-pmap-c-que)# bandwidth percent 80
switch(config-pmap-c-que)# class type queuing class-default
switch(config-pmap-c-que)# bandwidth percent 20
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input fcoe-80
switch(config-sys-qos)# service-policy type queuing output fcoe-80

switch(config)# policy-map type queuing fcoe-80
switch(config-pmap-que)# class type queuing class-fcoe
switch(config-pmap-c-que)# bandwidth percent 80
switch(config-pmap-c-que)# class type queuing class-default
switch(config-pmap-c-que)# bandwidth percent 20
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input fcoe-80
switch(config-sys-qos)# service-policy type queuing output fcoe-80
switch(config-sys-qos)# exit

```

10.4 STORAGE SYSTEM SETUP

Enabling the FC protocol and physically connecting the unified target adapters to a switch are the two steps required to set up the storage system. The connectivity between the initiator and the target interfaces can be confirmed with the `fcp show initiator` command. The WWPN of each initiator port on the server should be visible in the list of connected initiators.

```

Initiators connected on adapter 3b:
Portname          Group
-----          -
20:xx:xx:xx:xx:xx:xx:xx
20:xx:xx:xx:xx:xx:xx:xx

Initiators connected on adapter 5b:
Portname          Group
-----          -

```

```
20 : xx : xx : xx : xx : xx : xx : xx
20 : xx : xx : xx : xx : xx : xx : xx
```

10.5 SERVER SETUP

As it is for the storage system, only a few setup steps are required for the server after zoning is complete. Enabling FCoE requires one prerequisite: When running the driver update program, select the Data Center Bridging/FCoE option at the Setup Options screen before completing the driver update.

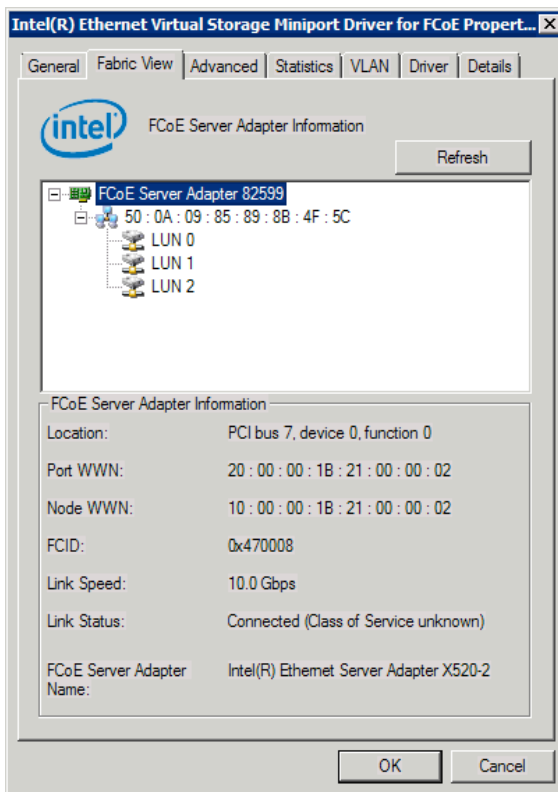
To view the visible targets to a given initiator:

1. Open the interface in Diagnostics, Device Manager, Storage Controllers.
2. Right-click the interface in question.
3. Select Properties.

The Fabric View pane lists both the local initiator WWPN and the WWPNs visible in the fabric.

This can also be useful for mapping a given physical interface to a given WWPN when creating FC fabric zones on the switch.

Figure 3) Fabric view.



When using SnapDrive for LUN provisioning, the steps are identical for a server attached to the storage system by using FCoE and a storage system attached by using iSCSI. The only difference is that when the initiator is selected to establish the connection, an FCoE initiator is selected by its WWPN instead of selecting an iSCSI initiator by its iSCSI qualified name (IQN).

11 RESULTS AND VALIDATION

First, we tested the speed capabilities of the Unified Connect adapters on the storage system, the Nexus switch, and the Intel X520 Ethernet adapters. We tested the speed by running I/O tests with IOMeter using block protocols (iSCSI and FCoE). Then, we demonstrated Unified Connect capabilities by addressing the storage system.

Next, we performed more functional tests relevant to an I/O-intensive application, Microsoft SQL Server 2008. This application requires both high throughput and low latency. First, we used the standard Microsoft I/O profiling and testing tool, SQLIOSim, on the LUNs created by SnapDrive. Then, we installed Microsoft SQL Server 2008 itself with default options using the same LUNs used as storage for the test databases, with approximately 4 million rows of sample data. Using NetApp SnapManager for SQL Server, we demonstrated the speed and simplicity of Snapshot-based backups and restores.

Finally, after removing the data, we used the SnapDrive Space Reclaimer functionality to reclaim space in the storage system aggregate that contained the test data.

11.1 IOMETER TESTS

For each listed test, we gathered the data by using the IOMeter with a target and the following settings:

- 8 outstanding I/Os on the target disk
- 1 worker configured per server core (for a total of 24, with Simultaneous Multithreading enabled)
- A maximum disk size of 10,000 sectors
- All I/O aligned on 4K boundaries to match the Data ONTAP 4K WAFL[®] (Write Anywhere File Layout) block size

We ran the following I/O tests at a variety of block sizes:

- 100% read, 100% sequential
- 100% read, 100% random
- 100% write, 100% sequential
- 100% write, 100% random
- 50% read, 100% sequential
- 50% read, 100% random
- 50% read, 50% random

The block size remained the primary factor throughout the tests, because the large amount of Flash Cache memory available in the FAS6280 storage system enables the data to be served mostly from cache, almost eliminating I/O wait time to fetch data from the disks.

In the tests that contained simultaneous read and write traffic, a throughput number greater than line speed can be observed. This is due to the duplex nature of an Ethernet medium (such as 10GbE Twinax, the one used in the lab environment) in which network adapters are able to both send and receive data simultaneously.

ISCSI RESULTS

Figures 4 through 10 show the iSCSI results. For simple read tests, both sequential and random, the server approached the line rate of 10Gb with a request size of 8KB; CPU utilization steadily decreased to approximately 7% for read requests over 128KB. The similarity of sequential and random results can be attributed to the large read cache size in the FAS6280. For write tests, the server approached a line rate at 16KB request sizes and achieved it at 32KB, with CPU utilization slightly lower than reads.

For the tests that were a mix of read and write, the full-duplex nature of 10GbE allowed the server to achieve near-line-rate speeds in both directions simultaneously.

Figure 4) iSCSI: 100% read, 100% sequential.

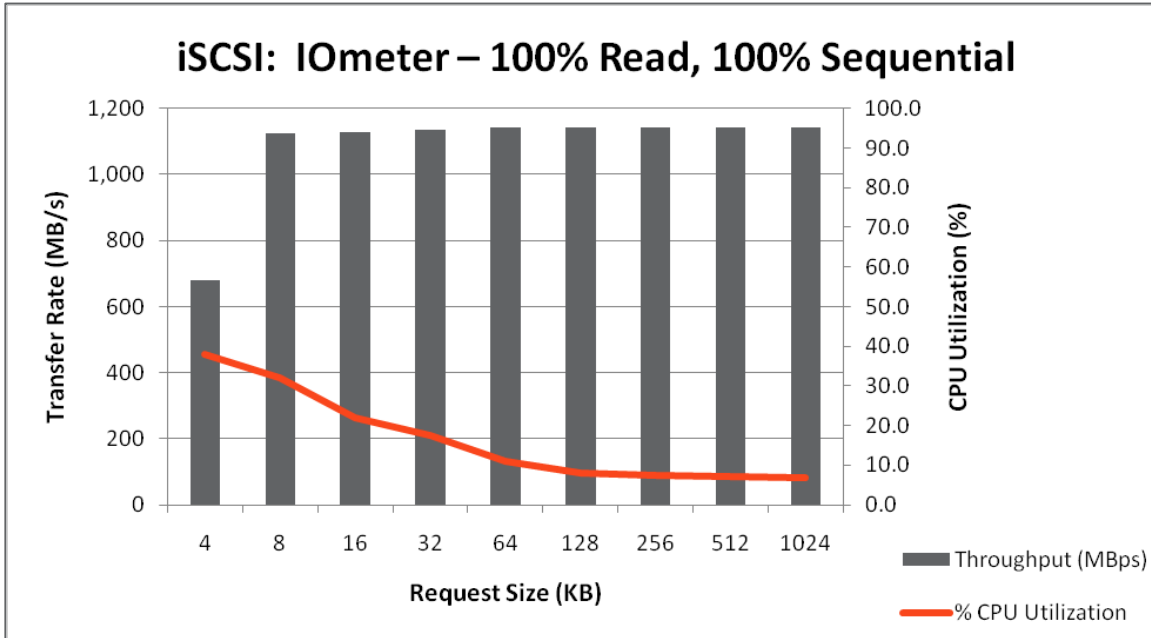


Figure 5) iSCSI: 100% write, 100% sequential.

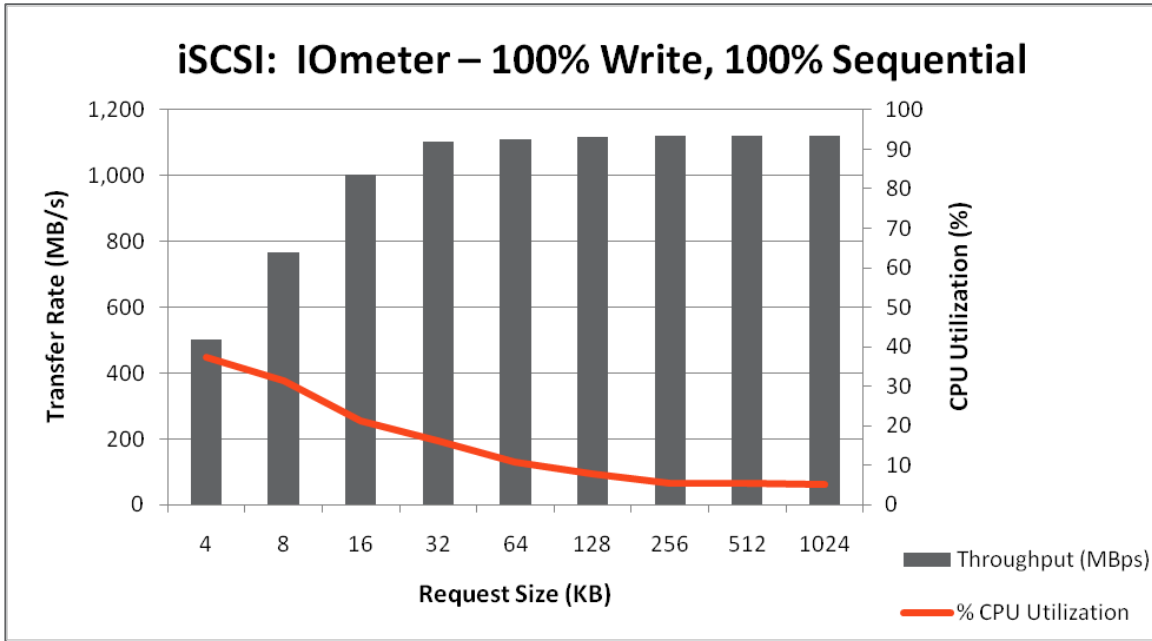


Figure 6) iSCSI: 100% read, 100% random.

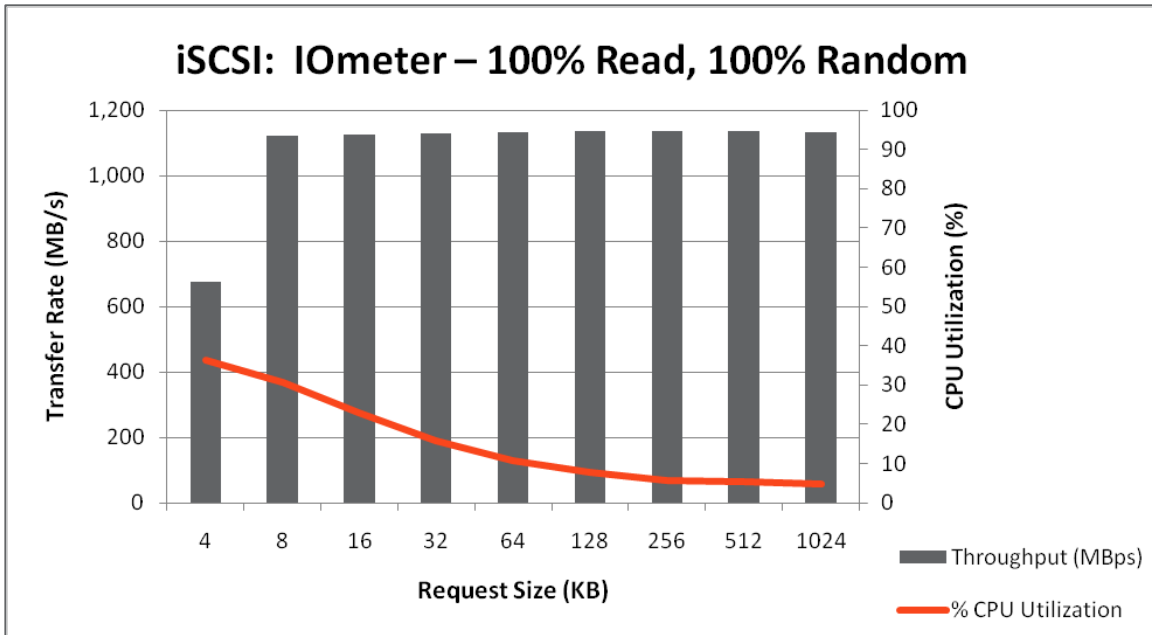


Figure 7) iSCSI: 100% write, 100% random.

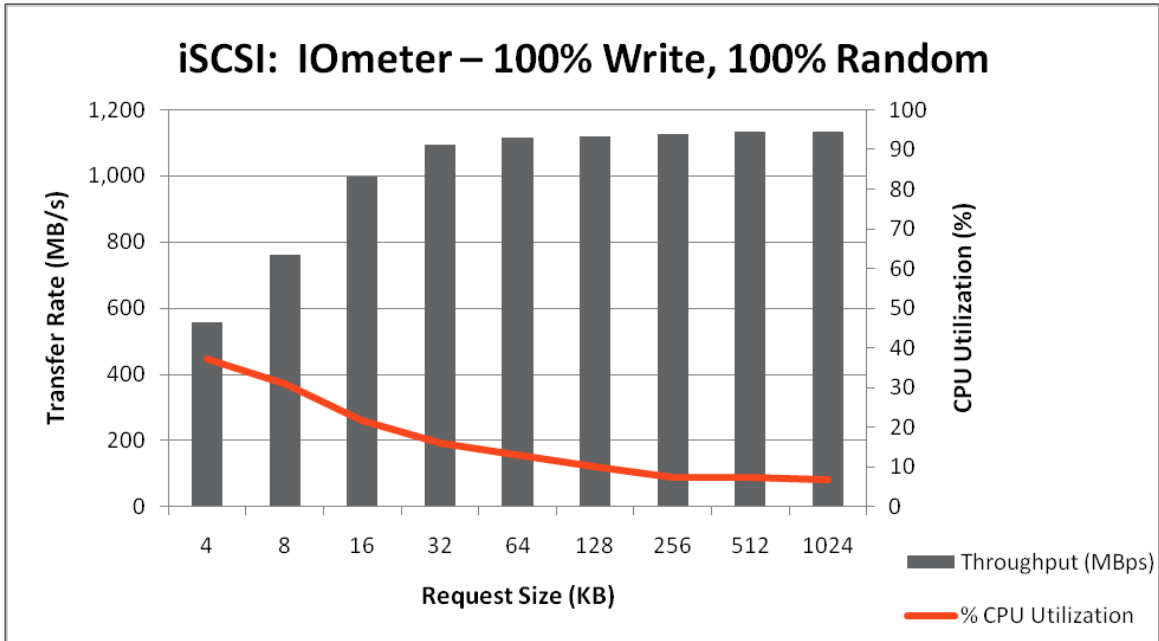


Figure 8) iSCSI: 50% read, 100% sequential.

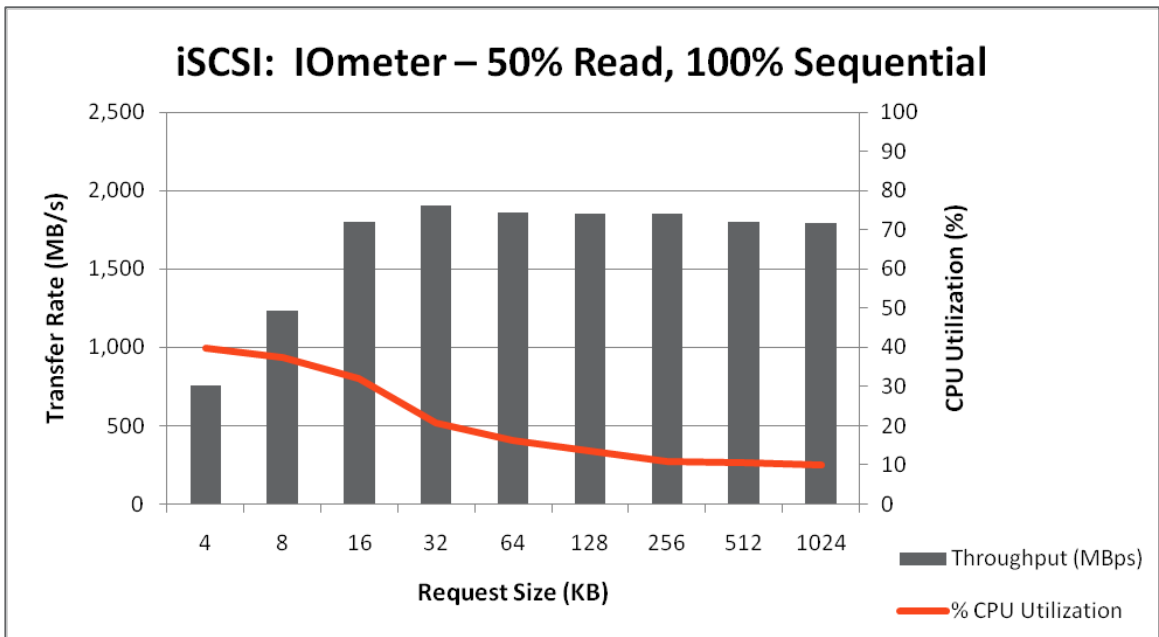


Figure 9) iSCSI: 50% read, 100% random.

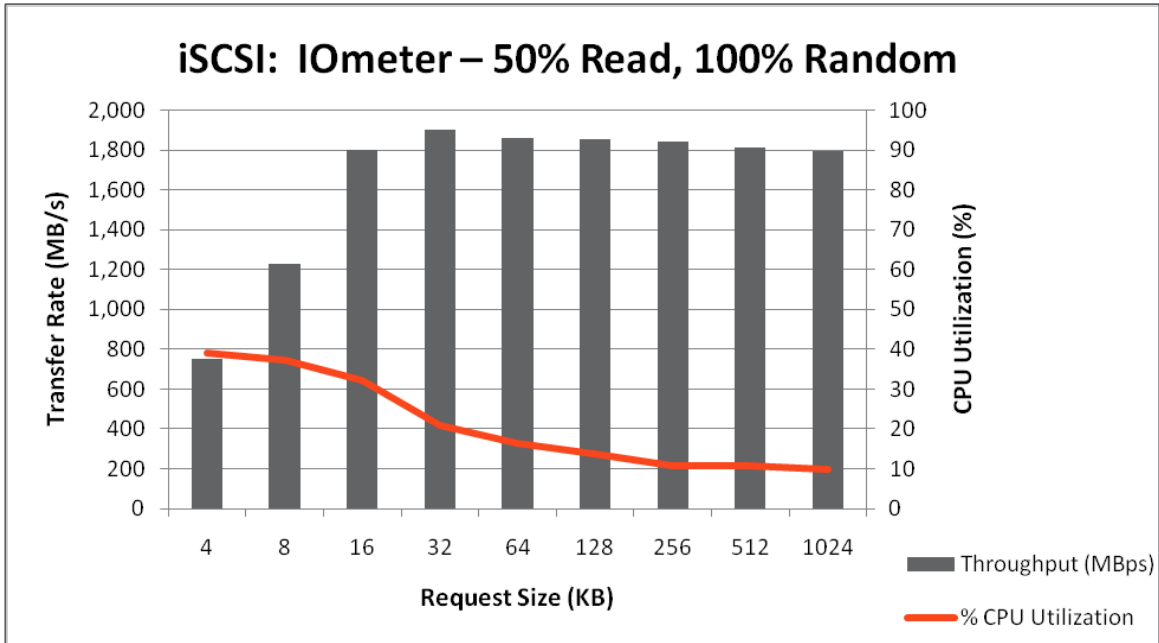
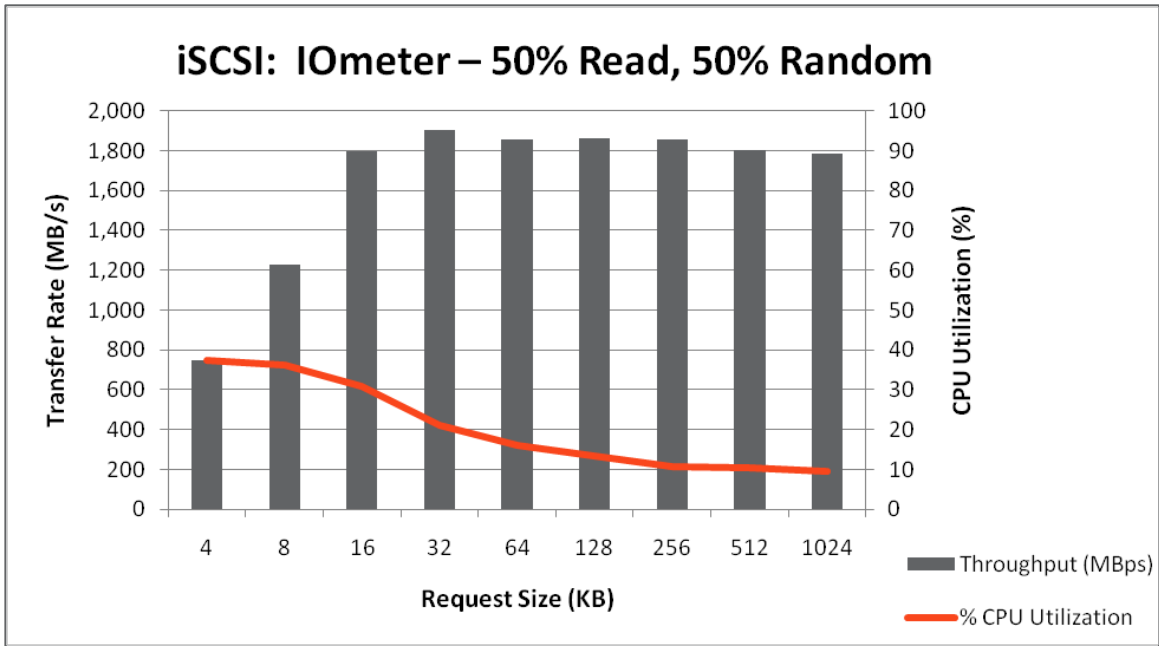


Figure 10) iSCSI: 50% read, 50% random.



FCOE RESULTS

Figures 11–17 show the FCoE results. They are remarkably similar to the iSCSI results, with the 10GbE line rate achieved at 8KB for pure reads and 16KB for pure writes, and CPU utilization quickly dropping as request sizes increase. Again, the difference between sequential and random workloads is minimized by the FAS6280's large cache. CPU utilization was slightly lower overall, but within approximately 5% of the iSCSI results.

Figure 11) FCoE: 100% read, 100% sequential.



Figure 12) FCoE: 100% write, 100% sequential.

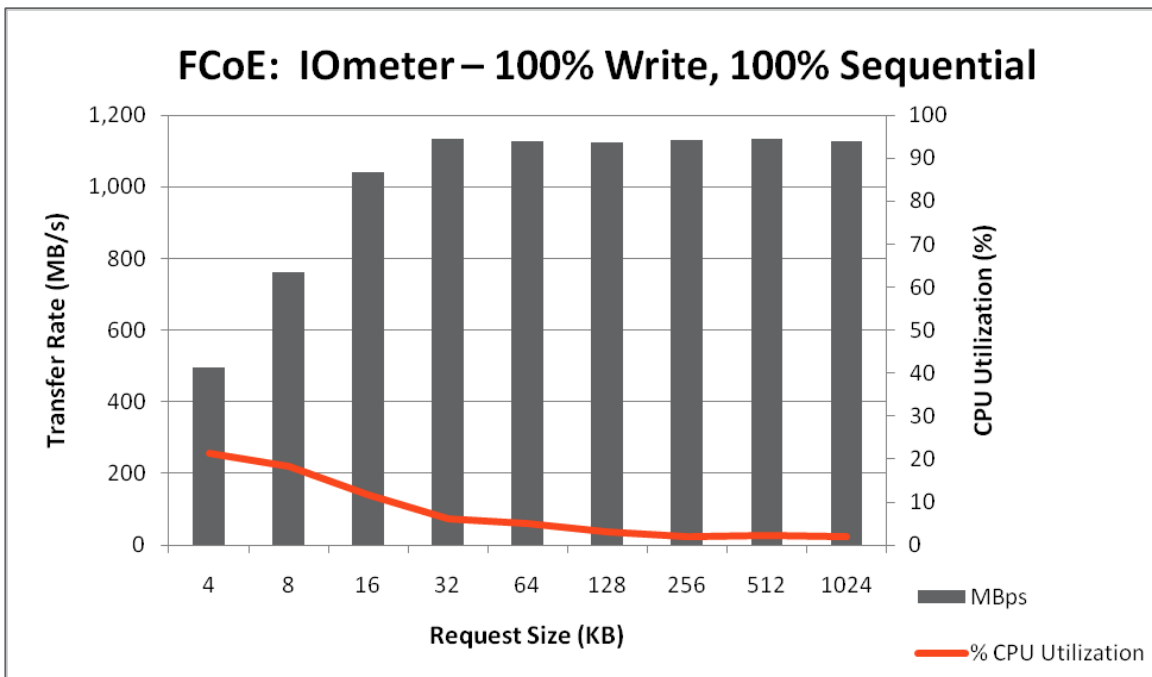


Figure 13) FCoE: 100% read, 100% random.

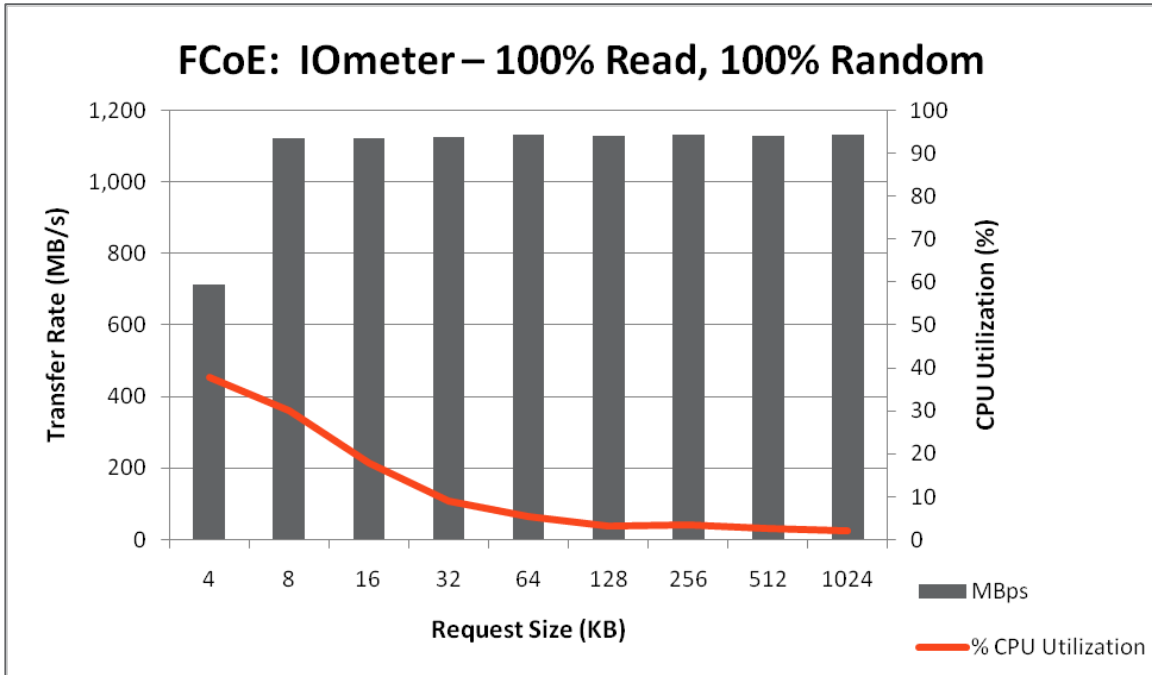


Figure 14) FCoE: 100% write, 100% random.



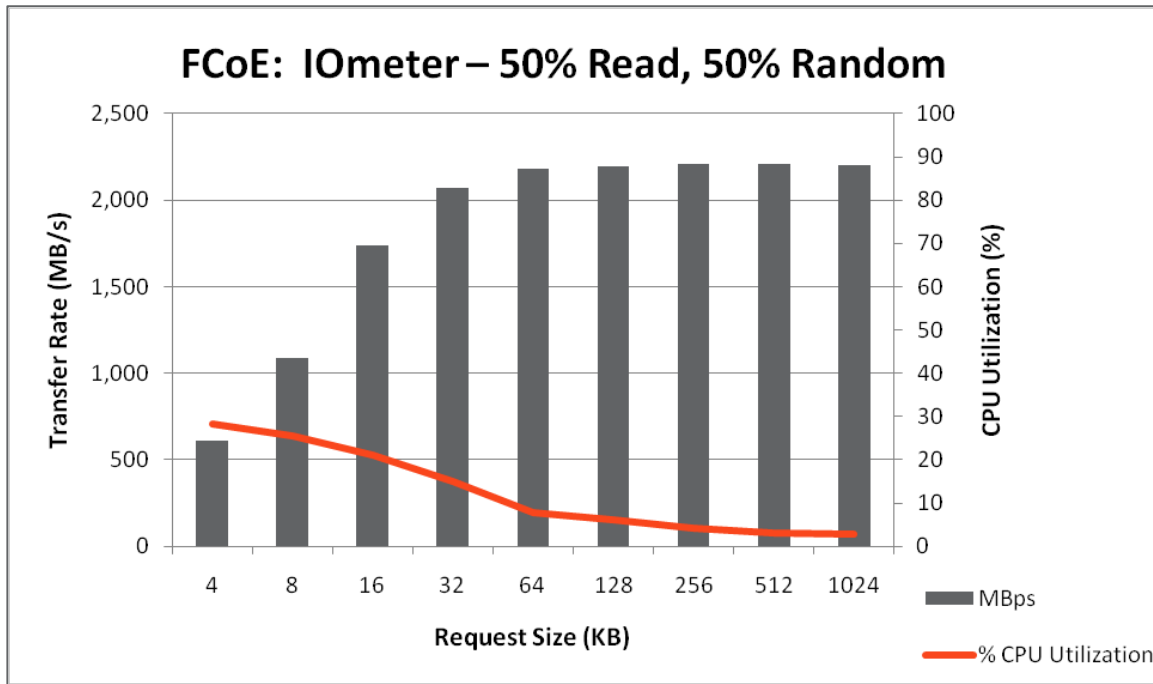
Figure 15) FCoE: 50% read, 100% sequential.



Figure 16) FCoE: 50% read, 100% random.



Figure 17) FCoE: 50% read, 50% random.



11.2 UNIFIED CONNECT (FCOE AND CIFS) TESTS

The unified target adapter and the X520 Ethernet adapter both permit simultaneous TCP/IP and FCoE traffic. Therefore, we ran the same I/O tests, this time with FCoE traffic accessing a LUN and CIFS traffic accessing a mapped drive simultaneously. Both used the same network interface and target adapter on the FAS6280. However, per best practice, the mapped drive and LUN were not contained by the same FlexVol volume on the storage system.

Both protocols were constrained by a class of service (CoS) dedicating 80% of the 10GbE line capacity to FC traffic and 20% to Ethernet traffic, as defined in the [Switch Setup](#) section for the FCoE configuration.

Even with a block and file storage protocol accessing the same port simultaneously, the FAS6280 and X520 Ethernet adapters had no difficulty maintaining the same line rate demonstrated in the single-protocol tests shown in the [IOmeter Tests](#) section. Networking performance was unchanged, with the CoS defining the 80/20 ratio of block-to-file traffic.

Figure 18) FCoE and CIFS: 100% read, 100% sequential.

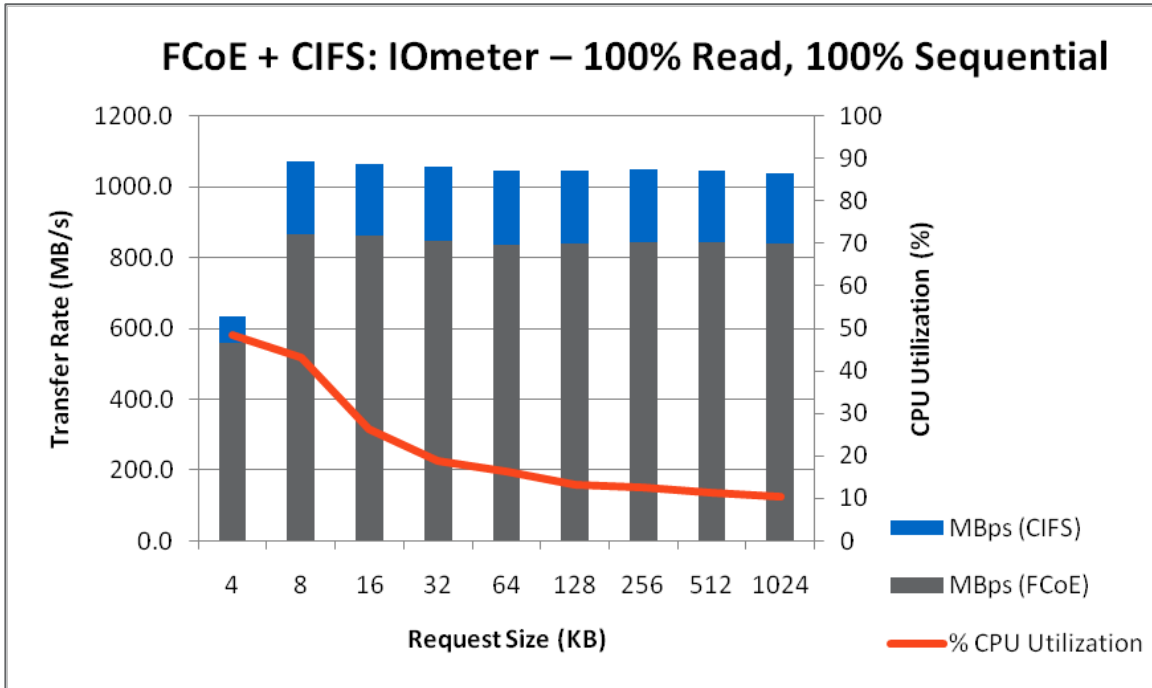


Figure 19) FCoE and CIFS: 100% write, 100% sequential.

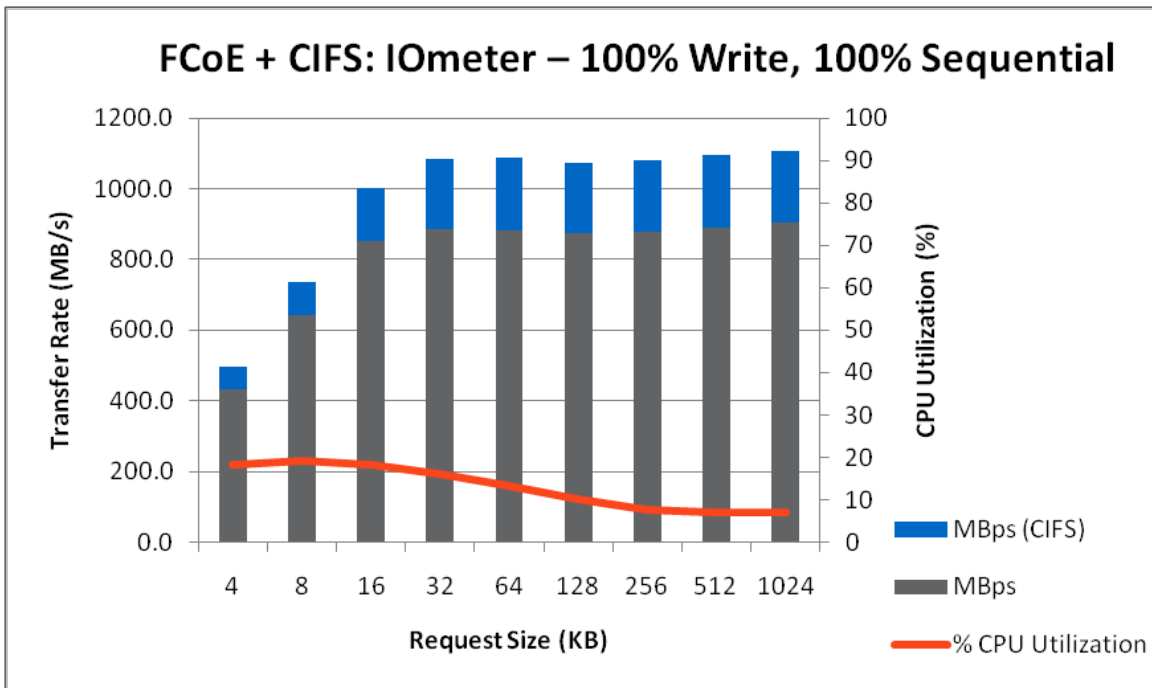


Figure 20) FCoE and CIFS: 100% read, 100% random.

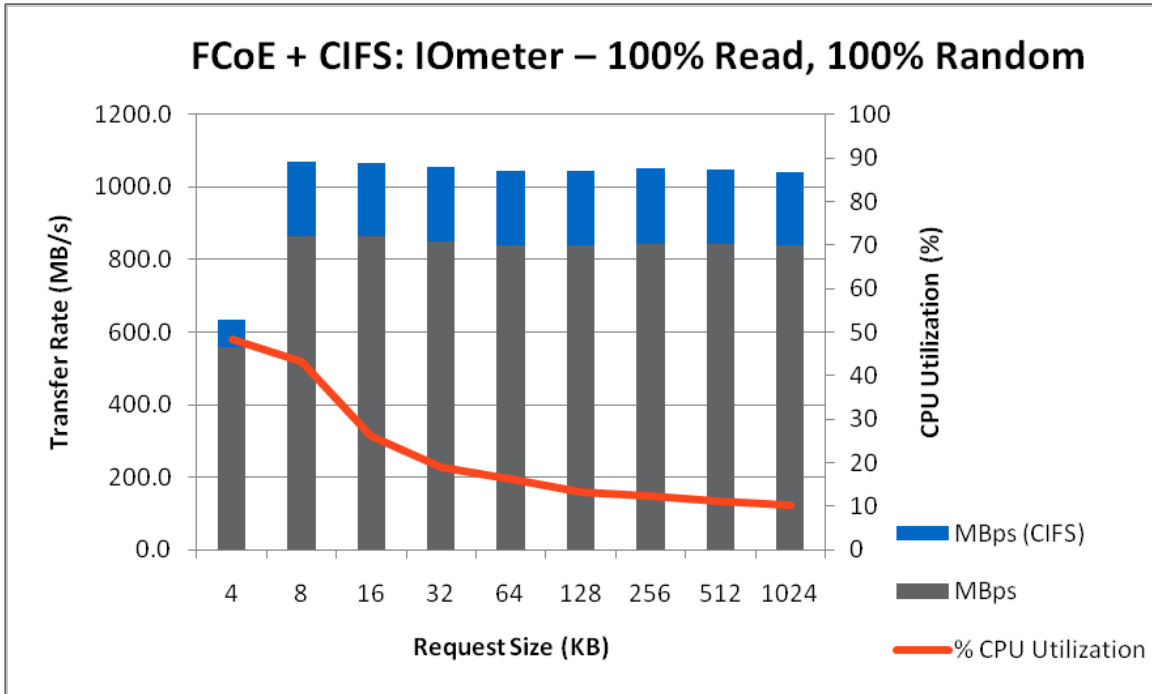


Figure 21) FCoE and CIFS: 100% write, 100% random.

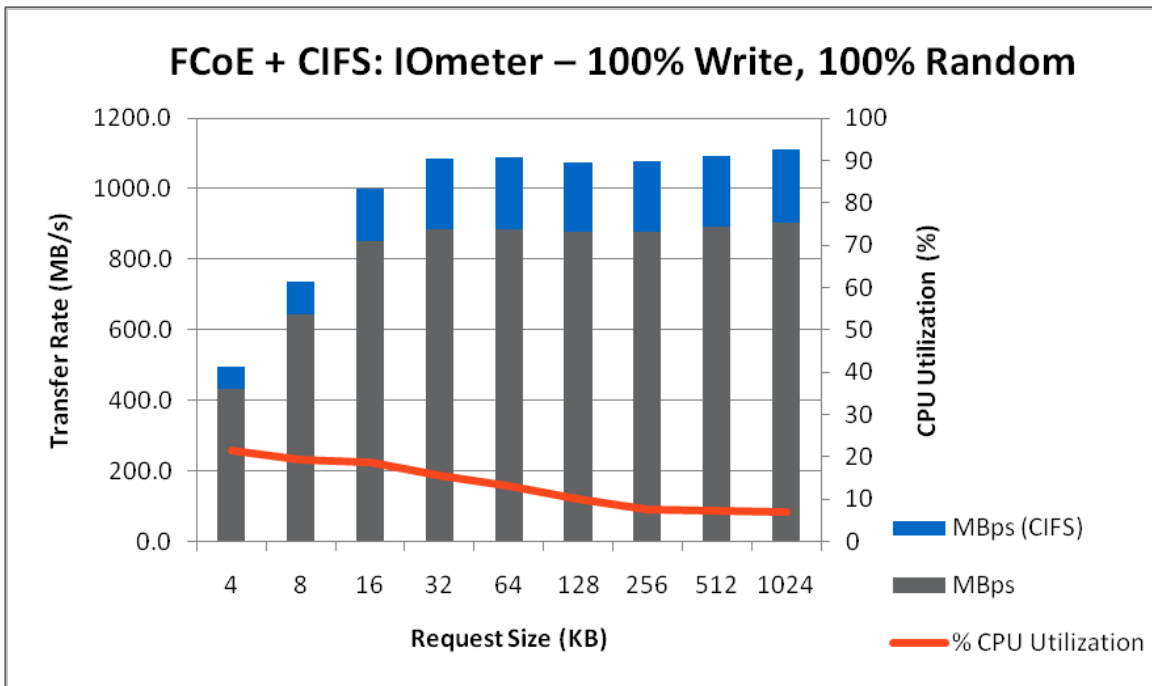


Figure 22) FCoE and CIFS: 50% read, 100% sequential.

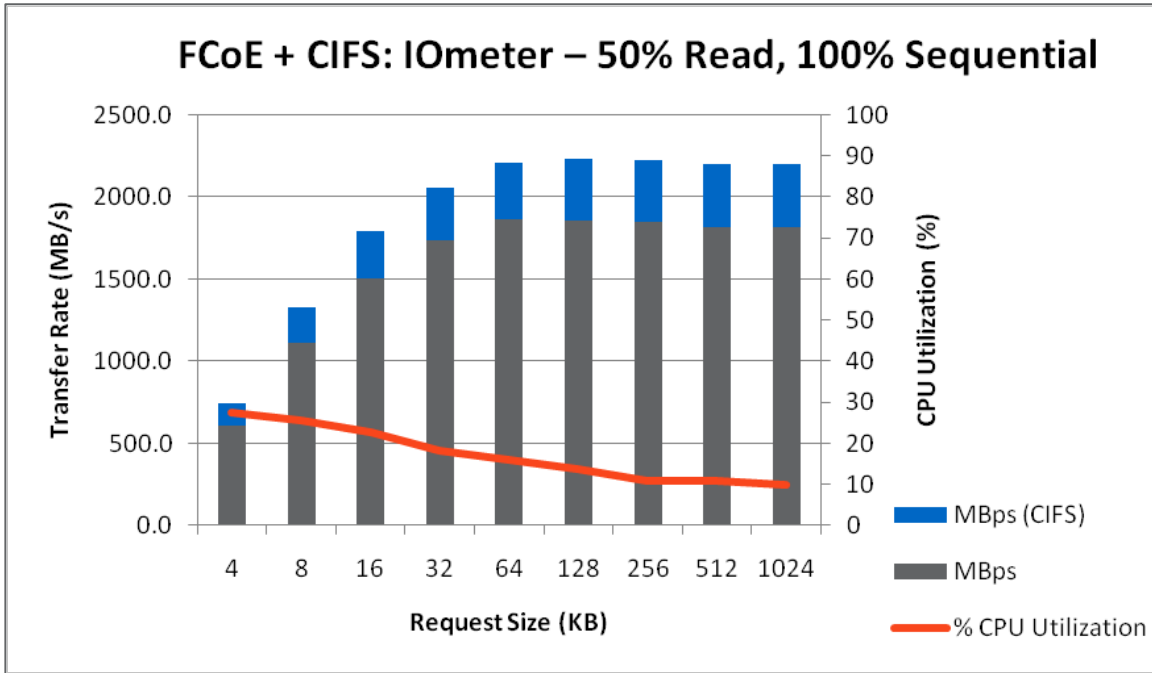


Figure 23) FCoE and CIFS: 50% read, 100% random.

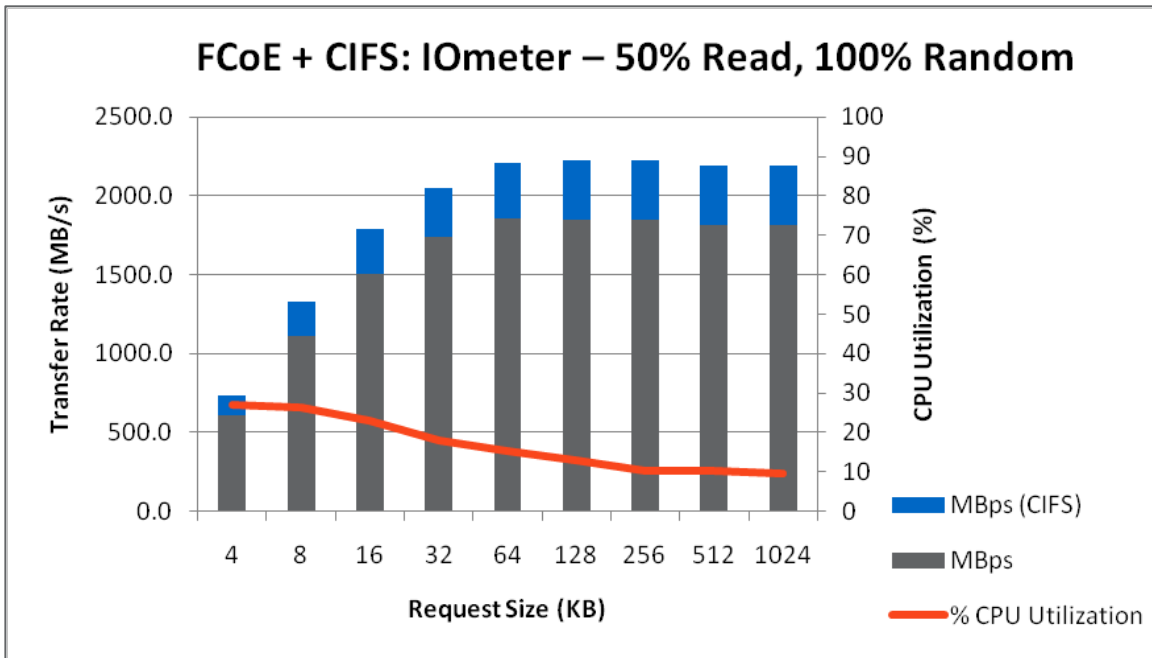
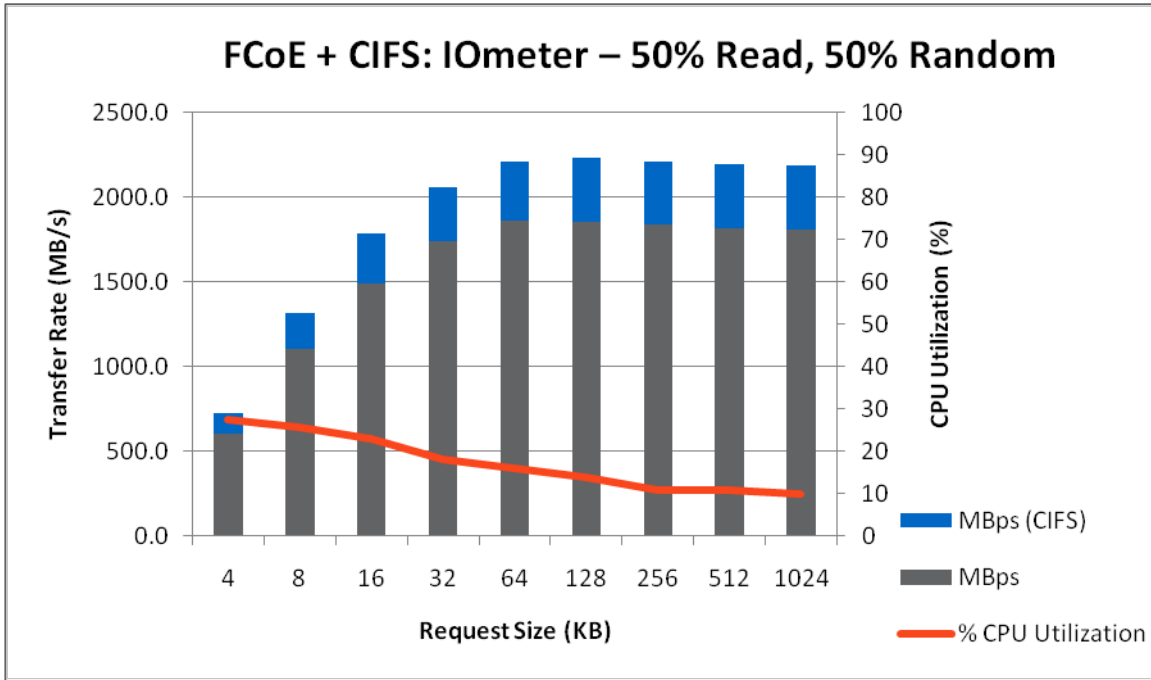


Figure 24) FCoE and CIFS: 50% read, 50% random.



11.3 SQLIOSIM TESTS

SQLIOSim is a Microsoft utility that simulates the I/O patterns of Microsoft SQL Server given a number of starting parameters. Because it is independent of the SQL Server engine and uses its own generated files on the target file system, it is only useful for getting a sense of the types of I/O loads that would be appropriate on a given server and storage system combination. It does not produce CPU or memory load on the server.

We ran three different SQLIOSim test configurations for both the FCoE-based and the iSCSI-based server systems:

- A basic test run—the SQLIOSim default configuration
- A test run without I/O throttling
- A test run consisting of mostly sequential writes without I/O throttling

In each case, the number of threads available to SQLIOSim is set to 24 to match the number of threads the server is able to execute simultaneously.

The results of these tests are shown in Table 1 and Table 2.

Table 1) Part 1 of test results.

	Prototype	File	Target I/O Duration (ms)	I/O Throttled	Concurrency	Average Latency (ms)	% of I/O Throttled
Standard	iSCSI	data1	100	1,816	587	0	0.017
	iSCSI	data2	100	2,915	290	0	0.032
	iSCSI	logfile	100	0	8	0	0.000
Unthrottled	iSCSI	data1	100	14,039	96	0	0.028
	iSCSI	data2	100	19,866	235	0	0.040
	iSCSI	logfile	100	0	8	0	0.000
Sequential Writes	iSCSI	data1	100	12,415	330	0	0.058
	iSCSI	data2	100	13,114	488	0	0.062
	iSCSI	logfile	100	N/A	N/A	N/A	N/A
Standard	FCoE	data1	100	53	290	0	0.001
	FCoE	data2	100	815	205	0	0.009
	FCoE	logfile	100	0	8	0	0.000
Unthrottled	FCoE	data1	100	277	308	0	0.000
	FCoE	data2	100	3,295	234	0	0.006
	FCoE	logfile	100	0	8	0	0.000
Sequential Writes	FCoE	data1	100	5,918	242	0	0.020
	FCoE	data2	100	5,035	214	0	0.017
	FCoE	logfile	100	N/A	N/A	N/A	N/A

Table 2) Part 2 of test results.

	Reads	Scatter Reads	Writes	Gather Writes	Total I/O Time (ms)
Standard	32,461	32,815	419	42,527	2,645,787
	26,137	26,053	496	38,839	2,312,030
	0	0	672,818	0	248,716
Unthrottled	148,514	135,708	3,369	213,754	26,475,360
	143,305	134,477	3,444	212,748	23,969,430
	0	0	506,196	0	640,021
Sequential Writes	960	17,915	28	195,783	40,264,805
	28	17,414	1	193,714	31,195,957
	N/A	N/A	N/A	N/A	N/A
Standard	30,527	31,069	455	39,835	2,045,869
	23,939	25,335	369	37,789	2,298,431
	0	0	343,325	0	95,537
Unthrottled	168,086	159,650	3,273	247,427	15,600,238

	Reads	Scatter Reads	Writes	Gather Writes	Total I/O Time (ms)
	161,493	158,735	3,311	247,348	19,073,704
	0	0	605,230	0	366,230
Sequential Writes	1,045	23,712	48	266,790	18,462,796
	46	22,849	0	266,748	30,811,725
	N/A	N/A	N/A	N/A	N/A

To summarize, both iSCSI and FCoE provided a roughly equivalent performance when tested by SQLIOSim. The test that was designed to measure large sequential writes produced slightly fewer overall throttled I/Os. These tests are not exhaustive and are not an indication of absolute performance for Microsoft SQL Server. Rather, they indicate that both of the block-based protocols can perform well and at a comparable level on a standard benchmark for SQL I/O patterns.

11.4 SPACE RECLAIMER

A primary storage efficiency advantage of using Microsoft Windows 2008R2, SnapDrive, and a NetApp storage system is the Space Reclaimer functionality.

With thin provisioning of LUNs, data is not preallocated in its containing FlexVol volume or aggregate; it is only removed from the pool of storage when the data is actually written.

In the past, the storage server did not have any visibility into the server file system and could not tell when a block inside that file system was marked as deleted. Thin provisioning was a one-way process as LUNs gradually filled with data, even if utilization of the server's file system returned to a lower level.

Space Reclaimer permits a storage or server administrator to reclaim deleted blocks from a thin-provisioned LUN by making the storage system aware of those blocks after a file system scan. This can increase storage efficiency significantly when dealing with file systems in which utilization is transient or likely to fluctuate.

To start using Space Reclaimer, simply select a disk in SnapDrive and then start Space Reclaimer from the Action menu.

Note: Space Reclaimer requires setting the LUNs with data that need to be reclaimed to No Reserve so that Data ONTAP does not automatically count their entire size against the free space remaining in their containing volumes.

The following utilization figures were obtained from the aggregate containing the test data while the tests were underway (excerpted from the `aggr show_space aggr1` command).

Volume	Allocated	Used	Guarantee
sql_fcoe_data	38278552KB	26380744KB	none
sql_iscsi_data	32525948KB	20635884KB	none
sql_fcoe_log	13545748KB	12414480KB	none
sql_iscsi_log	13383296KB	12262592KB	none

The following utilization figures were obtained after space was freed post-testing and Space Reclaimer was run:

Volume	Allocated	Used	Guarantee
sql_fcoe_data	2285719KB	1730751KB	none
sql_iscsi_data	2694091KB	1971914KB	none
sql_fcoe_log	560291KB	4449181KB	none
sql_iscsi_log	401854KB	12311868KB	none

Note: These utilization numbers only demonstrate the ability of Space Reclaimer to deallocate space used in a containing volume and aggregate. The amount of space reclaimed is entirely dependent on the number of files or blocks deleted on the server file system. Additionally, file system metadata on the server and the volume metadata on the storage system consume some amount of space even if the file system on the server is empty.

11.5 SNAPMANAGER FOR MICROSOFT SQL SERVER

SnapManager for Microsoft SQL Server integrates with both NetApp SnapDrive software and SQL Server to simplify and streamline backups.

After quiescing the database to be backed up, SnapManager instructs the SnapDrive service to create new, read-only Snapshot copies of the FlexVol volume where the database is stored on a NetApp storage server, capturing the master data file, any associated secondary data files, and the transaction log.

This backup method increases storage efficiency by making use of the WAFL file system and Snapshot capabilities to reduce the amount of storage required for backups. Data that remains unaltered since the last Snapshot copy consumes no space within the Snapshot copies. Because only the changes are captured, it's necessary to provision for the expected delta of change, not for multiple whole copies of data, so backups can be taken more frequently and retained longer.

SnapManager also handles the validation of backed-up data. In the simplest configuration, like the one used here, it instructs the SnapDrive service to connect to the Snapshot copy containing the backed-up data and then use SQL Server to run a validation check.

This process can be streamlined even further by using SnapManager. First, SnapDrive instructs the backing storage system to replicate the newly created backup data to a secondary storage system by using NetApp SnapVault® or NetApp SnapMirror®. Because these replication methods are Snapshot-based, the same storage efficiency space savings are retained. Furthermore, by communicating with another instance of SnapManager, a separate SQL Server can be instructed to handle the task of validating the backed-up database.

We tested both the backup and restore functionality after loading a local database on each server using a 2GB LUN with 4 tables and approximately 4 million lines of test data.

The following steps and backup logs show how SnapManager works through the backup procedure:

1. Determine which storage system-backed disks require the creation of a new Snapshot copy.
2. Halt all database I/O.
3. Create Snapshot copies on the volumes containing the database LUNs.
4. Record backup metadata in the SNAPINFO directory to maintain the catalog of backups.
5. Mount the Snapshot copies containing the backed-up LUN to the local host and verify their contents.
6. Unmount the Snapshot copies and report the backup results.

BACKUP LOG (EXCERPTED)

```
Backup Database1: NETAPPEPI02 : iSCSI-Test
Full database backup

Verify snapshot after backup(run DBCC) on NETAPPEPI02\ISCSI: Yes

*** SnapManager for SQL Server Report

Backup Time Stamp: 10-19-2010_14.09.06

Backup Database 1: NETAPPEPI02 : iSCSI-Test
```

```
Backup Database Group #1:
  NETAPPEPI02 : iSCSI-Test

Backup job based on volume: #1
Sub Job: #1 of Job#1
  1 - NETAPPEPI02 : iSCSI-Test

**** BACKUP JOB **** #1

Backup Time Stamp: 10-19-2010_14.09.06

**** Backup Group [#1] of Job#1
Backup Database List:
1 - NETAPPEPI02 : iSCSI-Test

Getting SnapInfo directory info...
1 - Database [iSCSI-Test] of [NETAPPEPI02] SnapInfo directory located at:
l:\snapinfo\SQL__NETAPPEPI02\DB__iSCSI-Test

SnapInfo file Name:
l:\snapinfo\SQL__NETAPPEPI02\DB__iSCSI-Test\FG__\10-19-
2010_14.09.06\SnapInfo__10-19-2010_14.09.06.sml

SQL Server is prepared to freeze the database now [NETAPPEPI02 : iSCSI-
Test]...

SNAPSHOT DATABASE

Preparing Snapshot to Drive:
LUN [M], LUN [L]
Snapshot Name: sqlsnap__netappepi02_10-19-2010_14.09.06
Begin to create snapshot...

SnapDrive is ready to create snapshot.
Create Snapshot Now [NETAPPEPI02 : iSCSI-Test]...

All VDI backup threads completed snapshot preparation and databases IO are
frozen for snapshot.

SnapManager are ready to create snapshot...
Snapshot virtual disk [M, L] for database [iSCSI-Test] successfully
completed.

TRANSACTION LOG BACKUP AFTER FULL BACKUP [#1]

Transaction log backup location for database [iSCSI-Test]:
l:\snapinfo\SQL__NETAPPEPI02\DB__iSCSI-Test\LogBackup\10-19-
2010_14.09.06__iSCSI-Test.TRB

Running verification from local server....
Mounting Snapshot [sqlsnap__netappepi02_10-19-2010_14.09.06] for LUN [M] of
Computer [NETAPPEPI02]

Mount point directory [C:\Program Files\NetApp\SnapManager for SQL
Server\SnapMgrMountPoint]
```

```
Mounting Snapshot [sqlsnap__netappepi02_10-19-2010_14.09.06] for LUN [L] of
Computer [NETAPPEPI02]
```

```
Mount point directory [C:\Program Files\NetApp\SnapManager for SQL
Server\SnapMgrMountPoint]
```

```
Attaching database [iSCSI-Test]...
```

```
Database attached.
```

```
DBCC CHECKDB (N'iSCSI-Test') WITH NO_INFOMSGS, PHYSICAL_ONLY
```

```
DBCC CHECKDB completed successfully.
```

```
Detaching database [iSCSI-Test].
```

```
Dismounting LUN C:\Program Files\NetApp\SnapManager for SQL
Server\SnapMgrMountPoint\MPDisk001 of SnapShot [sqlsnap__netappepi02_10-19-
2010_14.09.06]...
```

```
Server\SnapMgrMountPoint\MPDisk002 of SnapShot [sqlsnap__netappepi02_10-19-
Dismounting LUN C:\Program Files\NetApp\SnapManager for SQL
Server\SnapMgrMountPoint\MPDisk002 of SnapShot [sqlsnap__netappepi02_10-19-
2010_14.09.06]...
```

```
**** FULL DATABASE BACKUP RESULT SUMMARY #1 ****
```

```
Backup Time: 10-19-2010_14.09.06
```

```
Backup Group [#1]:
```

```
#1 : [NETAPPEPI02 - iSCSI-Test] : OK
```

```
*** SNAPMANAGER BACKUP JOB ENDED AT: [10-19-2010 14.24.49]
```

```
Full database backup with verification completed successfully.
```

RESTORE LOG (EXCERPTED)

```
Restore Host: NETAPPEPI02
```

```
Restore as database: iSCSI-Test
```

```
Restore Full Backup Set: 10-19-2010_14.09.06
```

```
Leave database operational: Yes
```

```
*** PREPARING RESTORE
```

```
Verifying the selection of backup data set...
```

```
*** RETRIEVE SNAPMANAGER BACKUP INFORMATION
```

```
Querying backup database set SnapInfo file path...
```

```
Checking SnapRestore License on LUN M:\...
```

```
Checking SnapRestore License on LUN L:\...
```

```
Connecting to SQL Server [NETAPPEPI02\ISCSI] for restore validation...
```

```
*** SNAPSHOT LOG/DATABASE VIRTUAL DISK BEFORE RESTORE
```

```
Creating Snapshot of LUN(s):
```

```
LUN [M], [L]
```

```
Snapshot name: rstsnap__netappepi02_10-19-2010_15.07.55
```

```
Snapshot SnapInfo drive(s) successfully completed.
```



```

DETACH DATABASE TO BE RESTORED

Connecting to SQL Server [NETAPPEPI02] for database detach...
Detaching database [iSCSI-Test] before restore...

Restoring Snapshot Full Database Backup [iSCSI-Test] of [NETAPPEPI02]...

Checking SQL Server [NETAPPEPI02]...
Connecting to SQL Server...
Sending SQL command to instance NETAPPEPI02 for database iSCSI-Test : device
NTAPDEV__NETAPPEPI02_iSCSI-Test...
Receiving by server [NETAPPEPI02]...
RESTORE DATABASE [iSCSI-Test] FROM
VIRTUAL_DEVICE='NTAPDEV__NETAPPEPI02_iSCSI-Test' WITH SNAPSHOT, NORECOVERY
Waiting for SQLServer to respond...
Performing data transfer of [NETAPPEPI02 : iSCSI-Test]...
VDI SnapRestore the snapshot now [NETAPPEPI02 : iSCSI-Test]...

SnapRestore Database

Snapshot Name: sqlsnap__NETAPPEPI02_10-19-2010_14.09.06
Restoring LUN [M]...
LUN [M] was successfully restored.
Restoring LUN [L]...
LUN [L] was successfully restored.
Successfully completed data transfer.

TRANSACTION LOGS RESTORE

Checking Transaction Log Backup Restore...

Applying transaction log backup - 1...
[l:\snapinfo\SQL__NETAPPEPI02\DB__iSCSI-Test\LogBackup\10-19-
2010_14.09.06__iSCSI-Test.TRB]
Restoring as the last transaction log unit in a chain of log backup.
Database [NETAPPEPI02:iSCSI-Test] Percent: 100 percent processed.100
Database [NETAPPEPI02:iSCSI-Test] RESTORE LOG successfully processed 4 pages
in 0.030 seconds (0.813 MB/sec).
Transaction log restore completed.

**** RESTORE RESULT SUMMARY ****
Restore Database - #1 [iSCSI-Test]: OK
Restore completed successfully

*** SNAPMANAGER RESTORE JOB ENDED AT: [10-19-2010 15.17.36]

```

12 SUMMARY

In addition to the increased power offered by Intel Xeon-based servers and NetApp FAS6200 storage systems, converged networking can reduce the footprint, price, and complexity commonly associated with deploying high-speed centralized storage throughout a computing environment.

Furthermore, converged networking gives storage and system architects their choice of protocols for accessing storage, and, when deployed on a 10GbE network, both FCoE and iSCSI provide uncompromisingly low latency and high throughput. With software-driven Ethernet NICs increasingly

available in servers, administrators can react quickly and flexibly to changing requirements without the need for additional hardware.

13 REFERENCES

- [Intel Cloud Builders Guide: Cloud Design and Deployment on Intel Platforms](#)
- [TR-3441: Windows Multipathing Options with Data ONTAP—FCP and iSCSI](#)
- [TR-3800: Fibre Channel over Ethernet \(FCoE\) End-to-End Deployment Guide](#)

14 ACKNOWLEDGMENTS

We thank the following people for their help with this report:

- Bob Albers (Intel)
- Craig Pierce (Intel)
- Sreeram Sammeta (Intel)

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.



www.netapp.com

© 2011 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further faster, Data ONTAP, FlexVol, RAID-DP, SnapDrive, SnapManager, SnapMirror, Snapshot, SnapVault, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Cisco and Cisco Nexus are registered trademarks of Cisco Systems, Inc. Intel is a registered trademark and Xeon is a trademark of Intel Corporation. Microsoft, Windows, and SQL Server are registered trademarks of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-3894-0111