



NETAPP TECHNICAL REPORT

SAN BOOTING WITH WINDOWS SERVER 2003 AND 2008 OPERATING SYSTEMS

Toby Creek and Deepak S N, Netapp
March 2009 | TR-3376

ABSTRACT

This technical report explores the issues surrounding the booting of Microsoft® Windows® 2003 Server and Windows Server® 2008 operating systems from a NetApp FCP- or iSCSI-enabled storage system. Best practices, architecture, and feature-based recommendations are described to facilitate the construction of a system that meets performance and functionality expectations.

TABLE OF CONTENTS

1	PURPOSE AND SCOPE	3
2	REQUIREMENTS AND ASSUMPTIONS.....	3
3	TERMINOLOGY.....	3
4	BASICS OF THE BOOT PROCESS	4
5	CONFIGURATION FOR BOOTING FROM THE SAN.....	5
	5.1 INITIAL INSTALLATION.....	5
	5.2 MIGRATION OF AN EXISTING INSTALLATION FROM LOCAL DISK TO A SAN DEVICE	6
6	BEST PRACTICES RECOMMENDATIONS	6
	6.1 VOLUME LAYOUT	6
	6.2 OPERATING SYSTEM PAGEFILE PLACEMENT	6
	6.3 MICROSOFT CLUSTER SERVICES AND SCSI PORT DRIVERS	7
	6.4 GENERATING A SYSTEM BOOT IMAGE FOR PROVISIONING	8
	6.5 LUN CLONING	9
7	MEASURING THE BOOT PROCESS I/O PROFILE	10
	7.1 WINDOWS ENTERPRISE SERVER 2003, FCS.....	10
	7.2 WINDOWS ENTERPRISE SERVER 2008, FCS.....	11
	7.3 SUMMARY AND ANALYSIS.....	13
8	CONCLUSION	13
9	REFERENCES.....	13

1 PURPOSE AND SCOPE

This paper covers the issues that are relevant to SAN booting Windows 2003 Server and Windows Server 2008 operating systems from a NetApp FCP- or iSCSI-enabled storage system. Specifically, this paper addresses the following topics:

- Explanation of the IA32 (x86) and IA64 boot process
- Configuration instructions for enabling SAN booting capability
- Application of NetApp virtualization features to SAN booting
- Backup and recovery process
- Best practices for provisioning and performance
- Examination of boot-time throughput requirements

2 REQUIREMENTS AND ASSUMPTIONS

For the information and procedures presented in this paper to be useful, you should:

- Have at least intermediate Windows Server operating system administration skills
- Be familiar with the operation and manipulation of the IA32 and IA64 architecture BIOS
- Have at least intermediate NetApp storage system administrative knowledge
- Have physical and administrative access to the server
- Have administrative access to the NetApp storage system

In addition, the NetApp storage system must have the appropriate hardware and protocol licenses to perform the activities outlined. The system's host bus adapter must use a supported version of the firmware, BIOS, and driver from the NetApp SAN Support Matrix.

For clarity, administrative commands for the examples in this paper are performed at the server or storage system text-based console. Browser-based or graphical user interface procedures can be substituted as appropriate.

The material presented in this paper is relevant to Windows Server 2003 and Windows 2004 Server operating systems. Much of the discussion and many of the examples may also be applicable to other operating systems with minor modifications, particularly topics that are specific to NetApp features and technology.

3 TERMINOLOGY

- **BIOS:** Basic input/output system, for IA32 (x86) systems. The system that performs POST and initiates the boot process. The BIOS contains the INT13 device service routines as well as device configuration information and boot device preference.
- **EFI:** Extensible Firmware Interface, for IA64 systems. Although the EFI is roughly equivalent in function to the IA32 BIOS, a completely different construct than INT13 is used to load bootstrap code.

- **FCP:** Fibre Channel Protocol, as defined by ANSI X3.230-1994 (ISO 14165-1).
- **HBA:** Host bus adapter, usually from a PCI-based host bus to a storage networking interconnect.
- **Initiator:** The data consumer in a storage networking relationship.
- **INT13:** The device service routine in the IA32 BIOS that loads the initial bootstrap code from the boot device and transfers processor control to it.
- **iSCSI:** Internet Small Computer System Interface, as defined by IETF RFC 3720.
- **MBR:** The Master Boot Record, contained in the first sector of a bootable device.
- **POST:** Power-on self-test, a set of routines that perform diagnostics and initialize a host system before the boot process begins.
- **SCSI-3:** Small Computer System Interface, version 3; the SCSI standard where the protocol was separated from the transport.
- **SID:** System identifier, the unique identifier generated and used by Windows Server operating systems for authentication and authorization.
- **Target:** The data provider in a storage networking relationship.
- **WAFL®:** Write Anywhere File Layout, the virtualization and RAID engine present in all NetApp storage systems.
- **WWPN:** Fibre Channel World Wide Port Name, the address of an individual Fibre Channel interface port on an initiator or target.
- **WWNN:** Fibre Channel World Wide Node Name. A Fibre Channel initiator and target can have multiple WWPNs, but only one WWNN.

4 BASICS OF THE BOOT PROCESS

The boot process of the IA32 architecture has not changed significantly since the early days of the personal computer. Before the actual loading of the operating system from disk takes place, a preboot process is completed by the host BIOS routines.

The overall boot process is as follows:

A. Preboot

1. **Power-on self-test:** The BIOS initiates a diagnostic test of all hardware devices for which a routine exists. Devices for which the system BIOS does not have direct knowledge, such as installed HBAs, execute their own routines after the system tests are complete.
2. **Initialization:** The BIOS routines clear system memory and processor registers and initialize all devices.
3. **Set the boot device :** Although multiple devices may be bootable (the CD-ROM, a floppy disk drive, network adapter, storage HBA), only one can be the actual boot device. The BIOS determines the correct boot device order based on each device's "ready" status and the stored configuration.
4. **Load the boot sector:** The first sector of the boot device, which contains the MBR, is loaded. The MBR contains the address of the bootable partition on the disk, where the operating system resides.

B. Boot

1. **Boot loader phase:** The Windows Server operating systems rely on a file called Ntldr, which is the boot loader. At this point in the boot process, the IA32 process switches from real mode, in which the BIOS routines run, to protected mode. Protected mode

- enables access to memory above 1MB and provides virtual memory support with a flat address space.
2. Operating system selection: Ntldr relies on the boot.ini file to provide disk paths and other parameters to allow passing kernel parameters or continuing the boot process by using a different system path.
 3. Operating system boot: This phase is the remainder of the operating system boot process, where devices are scanned and enumerated, the kernel is loaded and initialized, and user-level processing is started.

The 64-bit x86 architectures follow the same general process, although the mechanisms are completely different. For a discussion of these differences, see the white paper “Boot from SAN in Windows Server 2003 and Windows 2008 Server” (link in the references section of this paper).

5 CONFIGURATION FOR BOOTING FROM THE SAN

5.1 INITIAL INSTALLATION

Once an HBA is installed and its boot BIOS is enabled, the configuration of the boot device can be performed. Each system’s BIOS configuration has specific features and procedures for setting the boot order, so these steps are presented as a general outline.

Follow the manufacturer’s recommendations for enabling the boot BIOS. If multiple cards are present in the system, the boot BIOS may be enabled on only one of them. If that card fails, manual reconfiguration must be performed to enable the system to boot from the other card.

For iSCSI, the general steps are as follows:

1. Configure the HBA’s interface IP address and netmask. Optionally, a default gateway can be set and the iSCSI nodename changed to match the requirements of the environment.
2. Configure a LUN to be used for the boot device on the storage system and map it to the initiator nodename, as defined in the previous step.
3. Enter the storage appliance target portal IP address into the BIOS to facilitate location of the boot LUN.
4. Verify connectivity by using the BIOS utilities to scan for available LUNs on the target. If LUNs are configured on the target but cannot be discovered, check LUN masking on the target and network configuration.
5. Select the LUN to be used for the boot device from those discovered.

For FCP, the steps are much the same:

1. The initiator’s World Wide Port Name is preconfigured but alterable. If replacing a failed card, it may be easier to change the WWPN to match that of the failed initiator.
2. Configure a LUN to be used for the boot device on the storage system and map it to the initiator nodename, as defined in the previous step.
3. Enter the storage system’s World Wide Name into the BIOS configuration.
4. Verify connectivity by using the BIOS utilities to scan for available LUNs on the target. If LUNs are configured on the target but cannot be discovered, check zoning configuration on the switch and LUN masking on the target.
5. Select the LUN to be used for the boot device from those discovered.

When these steps are completed, the operating system can be loaded. For the Windows Server operating systems, press the F6 key during the initial boot process from the installation media to allow additional mass storage drivers for the HBA to be installed. Without these drivers, the installation program cannot format the boot device and install the operating system files.

5.2 MIGRATION OF AN EXISTING INSTALLATION FROM LOCAL DISK TO A SAN DEVICE

You may need to migrate an existing operating system installation to a SAN boot device. This type of migration requires a solution to replicate the existing local disk to the SAN boot device. The built-in mirroring function in Windows Server operating systems can be used to accomplish this, but it is important to note these limitations:

- NetApp does not currently support some operations on dynamic disks, such as resizing, with or without the SnapDrive® software.
- Snapshot™ images of boot devices can only be taken with the system shut down. SnapDrive does not support Snapshot creation on the boot disk.
- Once a boot disk is converted to a dynamic disk, it is not possible to revert to a basic disk.

The basic process for migrating an existing installation is as follows:

1. Install the HBA to be used for SAN boot, if it is not already present in the system.
2. Install the drivers, firmware, and any boot BIOS code that is required by the HBA.
3. Provision a LUN from the NetApp storage system to use as the boot device. The LUN must be at least as large as the original disk.
4. Convert the local boot disk and the LUN to dynamic disks. Reboot the server to complete the conversion process on the local disk.
5. Add the LUN as a mirror to the existing local disk.
6. Shut down the server and configure the system and HBA BIOS to set the mirrored LUN as the boot device.
7. Optionally, remove the local disk from the system.

Other disk replication software or methods may be available for Windows Server platforms, but they are outside the scope of this paper.

6 BEST PRACTICES RECOMMENDATIONS

This section examines the use of Data ONTAP® virtualization features, as well as some potential issues.

6.1 VOLUME LAYOUT

Volumes containing boot LUNs should be separated from application data to preserve Snapshot data integrity and to prevent Snapshot locking when using LUN clones. Even though volumes containing boot LUNs may not require much physical disk space, give the volume enough spindles so that performance is not bound by disk activity. With Data ONTAP 7G and later, volumes with boot LUNs can be created on the same aggregate in which the data volumes reside to maximize storage utilization without sacrificing performance.

6.2 OPERATING SYSTEM PAGEFILE PLACEMENT

The operating system pagefile is where Windows writes seldom-used blocks from memory to disk to free physical memory. This operation is called paging. Administrators are often concerned about placing the operating system pagefile on a SAN storage system.

Potential issues with placing the pagefile on a SAN device are:

- Where systems share common resources on the SAN, such as disk spindles, switch bandwidth, or controller CPU and cache, heavy paging operations of one system can

affect storage system responsiveness for both operating system and application data for all connected systems.

- Depending on the device configuration, paging to a SAN device may be slower than paging to local storage. This is unlikely, because paging operations benefit from the write cache and multiple disk spindles available with enterprise-class SAN storage systems. These benefits far outweigh the latency induced by a storage networking transport, unless the storage is oversubscribed.
- Sensitivity to bus resets can cause systems to become unstable. Bus resets do not generally affect all systems connected to the SAN, because Microsoft has implemented a hierarchical reset handling mechanism in its STORport drivers for Windows Server 2003 to address this behavior.
- High latency during pagefile access can cause systems to perform very poorly or to crash with a STOP message (blue screen). The disk array should be monitored carefully to prevent oversubscription of the storage, resulting in high latency.

Some administrators who are concerned about paging performance may opt to keep the pagefile on a local disk while storing the operating system on a NetApp SAN. There are potential issues with this configuration as well:

- If the pagefile is moved to a drive other than the boot drive, system crash dumps are not written. This can be an issue when trying to debug operating system instability in the environment.
- If the local disk fails and is not mirrored, the system crashes and is not be able to boot until the problem is corrected.

To provide the optimal solution it is recommended to put paging files on both the local mirrored disk and the NetApp SAN disk. Distributing the page file in this manner will result in the majority of paging happening to the local page disk because paging will be done to the less active partition and also allowing the capture of the memory dmp file when the computer crashes.

In general, if the system is paging heavily, performance suffers whether the pagefile is on a SAN device or a local disk. The best way to address this problem is to add more physical memory to the system or to correct the condition that is causing severe paging. Currently, the costs of physical memory are such that a small investment can prevent paging and preserve the performance of the environment.

It is also possible to limit the pagefile size or to disable it completely to prevent SAN resource contention. If the pagefile is severely restricted or disabled to preserve performance, it is likely that application instability may result in cases where memory is fully utilized. This option is recommended only for servers that have enough physical memory to cover the anticipated maximum requirements of the application.

6.3 MICROSOFT CLUSTER SERVICES AND SCSI PORT DRIVERS

The Microsoft Cluster Service uses bus-level resets in its operation. Without the ability to isolate these resets from the boot device, installations that use the SCSIport driver with Microsoft Windows 2003 Server or Windows Server 2008 must utilize separate HBAs for the boot device and the shared cluster disks. In deployments where full redundancy is desired, a minimum of four HBAs is required for MPIO. In Fibre Channel implementations, NetApp strongly recommends that zoning be employed to separate the boot and shared cluster HBAs.

By deploying Microsoft Cluster Services on a Windows Server 2003 platform using STORport drivers, both the boot disks and shared cluster disks can be accessed through the same HBA. A registry entry is required to enable a single HBA to connect to both shared and nonshared disks in an MSCS environment. See “Server Clusters: Storage Area Networks - For Windows 2003 and Windows Server 2008” (link in the references section of this paper).

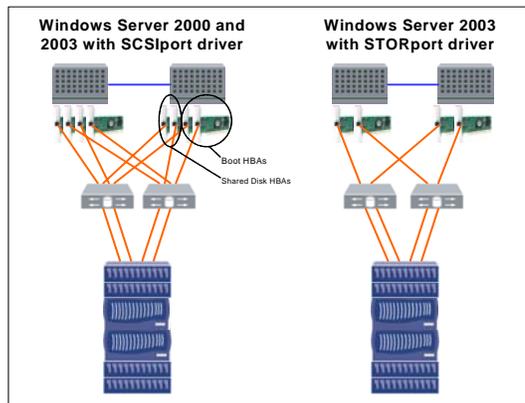


Figure 1) SCSIport versus STORport drivers in MSCS environments

6.4 GENERATING A SYSTEM BOOT IMAGE FOR PROVISIONING

With the LUN cloning capabilities of the NetApp Data ONTAP operating system, it is possible to rapidly make clones of existing LUNs for provisioning to new servers. With this functionality, a single boot image can be cloned to boot hundreds of servers, reducing the amount of time spent installing the operating system. Only the system personality, such as hostname and IP address, must be unique.

The following steps are required to generate a master system boot image:

1. Prepare an operating system installation, either new or migrated, on a boot LUN, as detailed in the previous sections.
2. Run the sysprep utility to remove all of the system personality information from the installation. Sysprep is included on the Windows Server 2003 CD. For Windows 2000 Server systems, the sysprep utility and documentation can be downloaded from the Microsoft Web site.
3. After sysprep is run, the system shuts down.
4. While the system is shut down, create a Snapshot copy on the NetApp storage system by using the command line interface or FilerView®.

```
filer> snap create bootvol golden
```

5. Clone the LUN, using either a LUN clone or a FlexClone® volume, and map it to the new host.

```
filer> lun clone create /vol/bootvol/anotherdisk.lun -b  
/vol/bootvol/masterdisk.lun golden  
filer> lun map /vol/bootvol/masterdisk.lun newserver  
lun map: auto-assigned newserver=0
```

6. At first boot from the cloned LUN, the system runs through an abbreviated setup process for final configuration. This includes setting the hostname and IP address. As part of this process, the system generates a new SID.

6.5 LUN CLONING

In environments such as massively parallel processing or compute farms where there is a high percentage of commonality in the operating system installations, administrators may attempt to reduce the amount of disk consumed by using LUN clones and a master image. This process is detailed in section 6.4. LUN clones use pointers into the Snapshot copy for unchanged blocks; only changed blocks consume additional space in the volume.

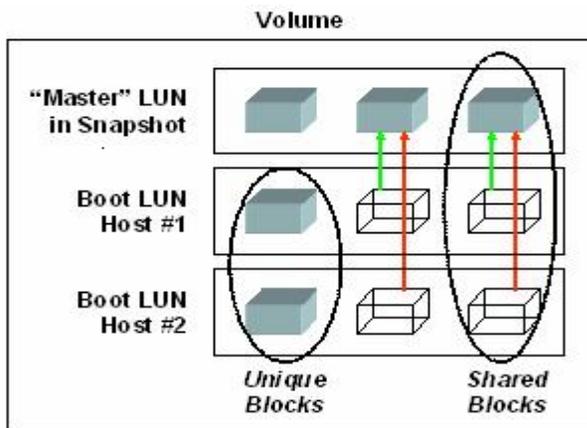


Figure 2) Example of block sharing.

Block sharing has two potential advantages over discrete boot LUNs:

- The amount of space consumed by the volume can be greatly reduced, especially because the number of hosts using the same master image grows.
- Because of the commonality of blocks in the LUNs, there is a higher probability that any shared block accessed in a LUN clone is already in cache, possibly improving performance.

There may be unexpected consequences in using this type of arrangement:

- Space reservations for cloned LUNs must be reduced or disabled to realize any space savings. Doing this could result in “write failed” SCSI sense codes being returned to the initiator if enough changed blocks are allocated to fill the volume. This is because not enough blocks are reserved to guarantee a complete overwrite of every block in each LUN clone. The most likely outcome of this event is that all systems with boot LUNs in that volume become unstable. Environments that do not use 100% space reservations should carefully monitor volume free space. Currently, the SnapDrive software suite does not support disabling space reservations for volumes in which it has configured LUNs.
- With unsplit cloned LUNs, changed blocks occupy new space in the volume. Intensive disk operations, such as disk defragmentation performed by the operating system, patches applied to the installation, or repeated pagefile grow and shrink operations can significantly and rapidly increase the space consumed.
- The space consumed by LUN clones can only increase over time. Because Data ONTAP does not examine blocks for commonality during writes, new blocks that contain identical data get discrete locations on disk.
- At least one Snapshot copy must be created to provide the master image for the hosts. When the first clone is created, the Snapshot copy that it was created from is locked. The Snapshot copy remains locked as long as the number of LUN clones created from it is nonzero. If additional Snapshot copies are created, they include the clones created previously, creating a condition in which it is very difficult to delete Snapshot copies without destroying LUNs first.
- Although very small, there is a performance impact associated with writes to cloned LUNs that does not affect split clones, because split clones do not share any blocks. During write operations to unsplit cloned LUNs, pointers in WAFL are manipulated and may have a performance impact. In testing, this impact has been negligible to the point that it is not measurable.

7 MEASURING THE BOOT PROCESS I/O PROFILE

Every storage network environment is different, so this paper does not try to comprehensively examine how to size a storage network. As mentioned previously, the boot process often requires much less of the storage than application data access does. For that reason, sizing estimates used for application data on servers using local disk are valid when used in an environment where the servers boot from the SAN.

This section discusses how to estimate boot time load. Sample data collected during testing is presented and serves to put the I/O load of the boot process in perspective with application data loads.

7.1 WINDOWS ENTERPRISE SERVER 2003, FCS

The data in the following table was collected using the same methodology and Server and Storage hardware

Pagefile	Boot time (sec)	Max iSCSI KB/sec	Total Read KB	Total Write KB	Read Ops	Write Ops
2.0GB initial, 4.0GB max	49	24928 (read)	82633	942	3710	126
Disabled	50	32954 (read)	78335	679	3792	115

The disk activity profile of the Windows Server 2003 operating system boot was very similar to that of the Windows 2000 Server, as would be expected. Somewhat surprisingly, however, there was much less write activity on boot. An initial 4-second burst totaling about 9.5MB was followed by 17 seconds of no disk activity at all. A 9-second burst containing the peak followed. This period contained 87% of the read activity, and tapered off to less than 1MB/sec for the duration of the OS load.

Disabling the pagefile made the disk activity bursts more intense, but shorter. The initial burst read about 4MB, followed by a 22-second idle period, then a 7-second burst, which read 67MB and tapered off to less than 1MB/sec for the duration of the boot

7.2 WINDOWS ENTERPRISE SERVER 2008, FCS

Windows Server 2008 can be installed in two types: full installation and core installation. Full installation supports GUI, and no roles are installed by default (for example, print, file, or DHCP). Core installation does not support any GUI, and it supports only command line and windows power shell; therefore requires lower memory and disk. The following testing was performed on Windows Server 2008, full installation type.

A few boot configuration changes are introduced in the Windows Server 2008. The major change is that boot configuration data (BCD) stores contain boot configuration parameters and control how the operating system is started in Windows Server 2008 operating systems. These parameters were previously in the Boot.ini file (in BIOS-based operating systems) or in the nonvolatile RAM (NVRAM) entries (in Extensible Firmware Interface-based operating systems). You can use the Bcdedit.exe command-line tool to affect the Windows code, which runs in the pre-operating system environment by adding, deleting, editing, and appending entries in the BCD store. Bcdedit.exe is located in the \Windows\System32 directory of the Windows 2008 active partition. BCD was created to provide an improved mechanism for describing boot configuration data. With the development of new firmware models (for example, the Extensible Firmware Interface), an extensible and interoperable interface was required to abstract the underlying firmware.

The following table shows the server and storage configurations that were used in gathering the results. The boot LUN was 30GB in size, created on aggregate with 9 ATA 7200 RPM drives on a flexible volume.

Server Physical Configuration

Server Layout	FC Card	iSCSI-enabled Gigabit Ethernet Card	Memory	Processor
IBM3655	Qlogic 2GB	Intel	13Gb	2.6GHz Opteron dual core

Storage Physical Configuration

Storage	RAID Type	No of Disks in Aggregate	Type of Disks	Volume Type
FS 3020	RAID-DP™	9	ATA – 7200 RPM	Flexible Volume

The Windows Server 2008 operating system was remotely installed on the NetApp storage using a Fibre Channel HBA card and an iSCSI software-enabled gigabit Ethernet adapter. Boot time was measured with and without pagefiles setting on the Windows Server 2008 over FC HBA and iSCSI software-enabled adapters.

FC HBA Remote Boot Results

Pagefile (GB)	Boot Time (sec)	Read (KB/s)	Write (KB/s)	Read (Ops)	Write (Ops)	Burst I/O (Mb/s)	Peak Time Occurred (Sec)
10 initial, 18 max	40	179301	1070	10235	189	30	36
Disabled	43	200268	4348	12725	754	41	35

iSCSI-Enabled Gigabit Ethernet Remote Boot Results

Pagefile (GB)	Boot Time (sec)	Read (KB/s)	Write (KB/s)	Read (Ops)	Write (Ops)	Burst IO (Mb/s)	Peak Time Occurred (Sec)
10 initial, 18 max	90	196239	1613	13676	798	33	62
Disabled	93	221801	4344	14870	814	46	63

- The Windows Server 2008 boot time on Fibre Channel was 50 times faster than on iSCSI software-enabled network adapter.
- During the reboot over FC or iSCSI there were more I/O read operations than write operations, which is true in the Windows Server 2003 boot phase also. No huge gap was seen in total read I/O and operations per second between FC and iSCSI software-enabled adapter during reboot. In Fibre Channel, initially between 10 and 15 seconds of remote boot there was 12MB of disk read activity followed by no disk activity. At peak 36 seconds there was a burst I/O of 30MB, followed by 80% of disk read activity on the iSCSI software-enabled adapter. In initial remote boot there was a total of 20MB disk read activity, followed by zero disk activity, and then at peak 62 seconds there was a burst I/O of 33MB, followed by 80% of disk read activity. This was observed with pagefile set on the Windows Server 2008.
- With pagefile not set on the Windows Server 2008, during FC remote boot during the initial 10 -to15 seconds, there was a total of 15MB of disk read activity, followed by no disk read activity. At peak 35 seconds there was a 41MB burst I/O followed by 80% of disk activity on the iSCSI software-enabled adapter. In the initial remote boot there was

total of 25MB disk read activity, followed by zero disk activity; and then at peak 63 second there was a burst I/O of 46MB, followed by 80% of disk read activity.

7.3 SUMMARY AND ANALYSIS

In all of the scenarios described in this report, once the boot process was complete and applications were started, steady state iSCSI activity to the boot drive was less than 2MB per minute. This was measured with no pagefile activity, because the operating system and sample applications did not exceed the available physical memory.

When profiling the system, it is important to take into account the application program loads during boot. In the data presented here, the operating system was installed without applications to isolate its boot I/O profile.

In both cases, operating system boot I/O loads were a fraction of the application steady state data access. Except in environments where frequent boots are anticipated, it is probable that boot does not present a load significant enough to be given great consideration in the environment, except possibly where I/O peaks absorb the available storage resources.

8 CONCLUSION

Booting systems from the storage area network offers significant advantages to administrators and to the organizations that they support. NetApp storage systems offer advanced virtualization features that greatly simplify provisioning. In most environments, application I/O is the primary motivator for sizing the system, rather than the variety of boot hosts.

9 REFERENCES

Configuring iSCSI SAN Booting with the QLogic QLA 4010/4010C HBA

<http://now.netapp.com/NOW/knowledge/docs/hba/iscsi/qlogic/pdfs/sanboot.pdf>

SAN Host Attach Kit 3.0 for Fibre Channel Protocol on Windows

<http://now.netapp.com/NOW/knowledge/docs/hba/win/relhbaw30/pdfs/setup.pdf>

iSCSI Host Support Kits for Windows

http://now.netapp.com/NOW/download/software/kit_iscsi/Windows/

FCP Host Support Kits for Windows

http://now.netapp.com/NOW/download/software/sanhost_win/Windows/

Snap Drive 5.0 for Fibre Channel and iSCSI Protocol on Windows

<http://now.netapp.com/NOW/knowledge/docs/snapdrive/relsnap50/pdfs/admin.pdf>

White Paper: Boot from SAN in Windows Server 2003 and Windows 2000 Server

<http://www.microsoft.com/windowsserversystem/storage/solutions/sanintegration/bootfromsaninwindows.mspx>

STORport in Windows Server 2003: Improving Manageability and Performance in Hardware RAID and Storage Area Networks

<http://www.microsoft.com/windowsserversystem/storage/technologies/storport/storportwp.mspx>

Server Clusters: Storage Area Networks - For Windows 2000 and Windows Server 2003

<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/clustering/starenet.mspx>

Microsoft Storage Technologies – Boot From SAN

<http://www.microsoft.com/windowsserversystem/storage/technologies/bootfromsan/default.mspx>

Support for Booting from a Storage Area Network (SAN)

<http://support.microsoft.com/default.aspx?scid=kb;en-us;305547>

Support for Booting from a iSCSI software enabled Network Adapters (iSAN)

<http://www.netapp.com/library/tr/3644.pdf>

iSCSI and Fiber Channel Configuration Guide

http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/QuickRef/fc_iscsi_config_guide.pdf