Technical Report

Performance Study of IBM DB2 9 on AIX 5L with NFS, iSCSI, and FCP Using IBM N Series or NetApp Storage System

Jawahar Lal and Roger Sanders, NetApp
Nailah Bissoon, Sunil Kamath, and Augie Mena, IBM
May 2009 | TR-3581

EXECUTIVE SUMMARY

This paper provides a performance comparison and tuning recommendations for three different transport protocols—FCP, iSCSI, and NFS, when running IBM DB2 9 and AIX 5L 5.3 TL-04 on a two-way IBM System p5 520 server using a FAS3050 NetApp storage system. An OLTP workload was used to study the performance, and the results show that the throughput achieved for NFS and iSCSI was within 7%, with FCP bearing an advantage over the other two protocols.

TABLE OF CONTENTS

1 INTRODUCTION

The increasing demand for information has resulted in the need for performance improvements to storage systems as well as new levels of performance for communications between server and storage. There are two main storage paradigms used to enable connections between a server and a storage system: network-attached storage (NAS) and storage area network (SAN).

NAS is data storage that is connected to a traditional Ethernet network and made accessible to clients on those networks via standard protocols. The most common NAS solution is the Network File System (NFS). NFS not only simplifies data transport but also reduces the management complexity of large storage installations.

SAN relies on a set of storage specific transport protocols—iSCSI and FCP. Since iSCSI uses TCP/IP as its transport for SCSI, information can be passed over existing IP-based host connections typically via Ethernet. FCP, in contrast, uses Fibre Channel interconnect between server and storage.

Currently, Ethernet bandwidth is eclipsing Fibre Channel, and the cost of Ethernet is becoming more affordable than the costs of Fibre Channel. One question that arises is how NFS and iSCSI perform in comparison to FCP. This paper attempts to compare the performance while running an OLTP workload with DB2 9 on AIX5L 5.3 TL-04 using iSCSI, FCP and NFS with an IBM® N5500 storage system.

2 ASSUMPTION

In order to take maximum benefit from this study, we assume that the reader of this paper has a fair understanding and knowledge of the following products and technologies:

- AIX 5L operating system
- Protocols such as FCP, iSCSI, and NFS
- DB2 9 Enterprise Server Edition (ESE) terminologies
- Data ONTAP®

3 SYSTEM CONFIGURATION

The data server used for this study was an IBM System p5 520 server running AIX 5L 5.3 TL04 with 8GB of physical RAM and two 1.5GHz processors running DB2 9 Enterprise Server Edition. For data storage, a NetApp® storage system with two nodes running Data ONTAP 7G was used. Two 1 Gigabit Ethernet network adapters established the NFS and iSCSI connections, and two 2GB FC-AL adapters were used for the FCP connection between the data server and storage system; this supported the bandwidth availability at the time. Figure 1 shows the high-level system architecture of the data server and the NetApp storage system.

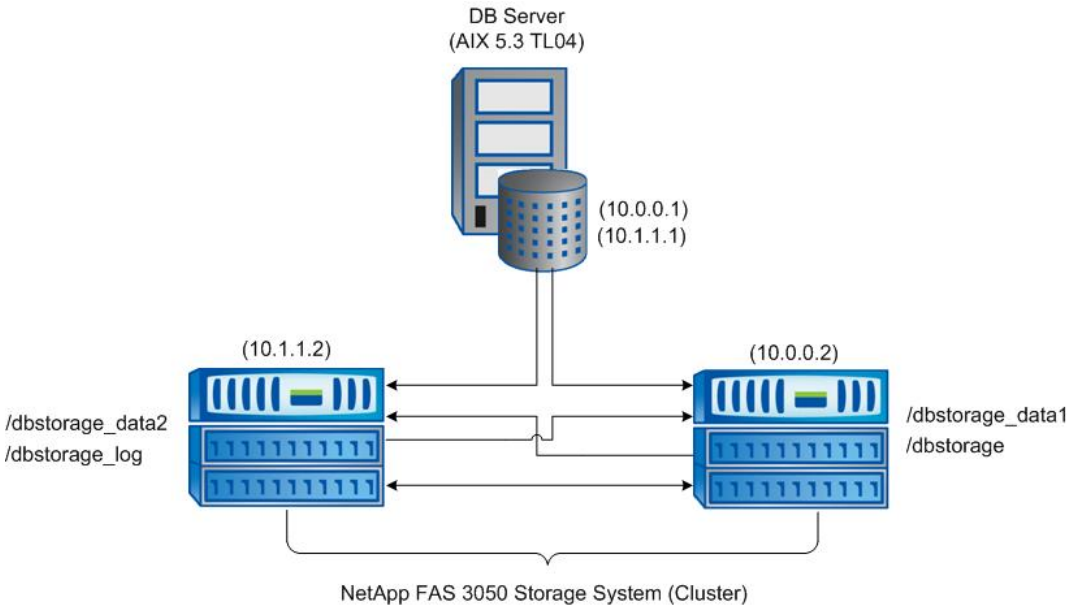

Figure 1) System architecture.

4 DATABASE TUNING

Our test environment consisted of a DB2 9 Enterprise Server Edition database running an Online Transaction Protocol (OLTP) workload. The size of the database was approximately 20GB. The tablespaces associated with the database were created with the NO FILE SYSTEM CACHING clause, which is available only on the IBM Enhanced Journaling File Systems (JFS2).

The operating system by default caches file data that is read from and written to disk. For OLTP workloads, caching at the file system level and in the DB2 buffer pools causes performance degradation due to the extra CPU cycles required to do double caching. DB2 uses the Concurrent I/O (CIO) feature to disable file system caching when either the CREATE TABLESPACE or ALTER TABLESPACE statement is used with the NO FILE SYSTEM CACHING clause. For more details on CIO, refer to "Improve Database Performance on File System Containers in IBM DB2 UDB Using Concurrent I/O on AIX" [6].

5 NETAPP STORAGE SYSTEM

Achieving efficient transaction response times depends not only on the database management system (DBMS), but also fast access to data as well as the transaction logs is crucial. NetApp storage systems offer efficient connectivity to the data server by supporting various transport protocols such as FCP, iSCSI, and NFS. The storage layout on the NetApp system storage is outlined below as well as some recommendations for optimizing the storage system for OLTP workloads.

5.1 STORAGE LAYOUT

Before a NetApp storage system can be used to store tablespace containers, it must first be configured, and the appropriate storage objects such as aggregates, flexible volumes, and LUNs need to be created. The steps required to configure a NetApp storage system for a DB2 environment are described in detail in the technical report entitled "DB2 9 for UNIX: Integrating with a NetApp Storage System" [2].

In our environment, two aggregates were created, one on each node of the clustered storage system. The aggregates were created using the following command:

```
create aggr0 -r 20 40@36g -t RAID_DP
```

In the above mentioned example, aggr0 is the name of the aggregate that was created over 40 disks, each 36GB in size. Double parity and a RAID group size of 20 were also used. Double Parity RAID (RAID-DP™) allows greater data protection in the event of double disk failures. Figure 2 illustrates the resulting aggregate that was created on the IBM N5500 system storage.

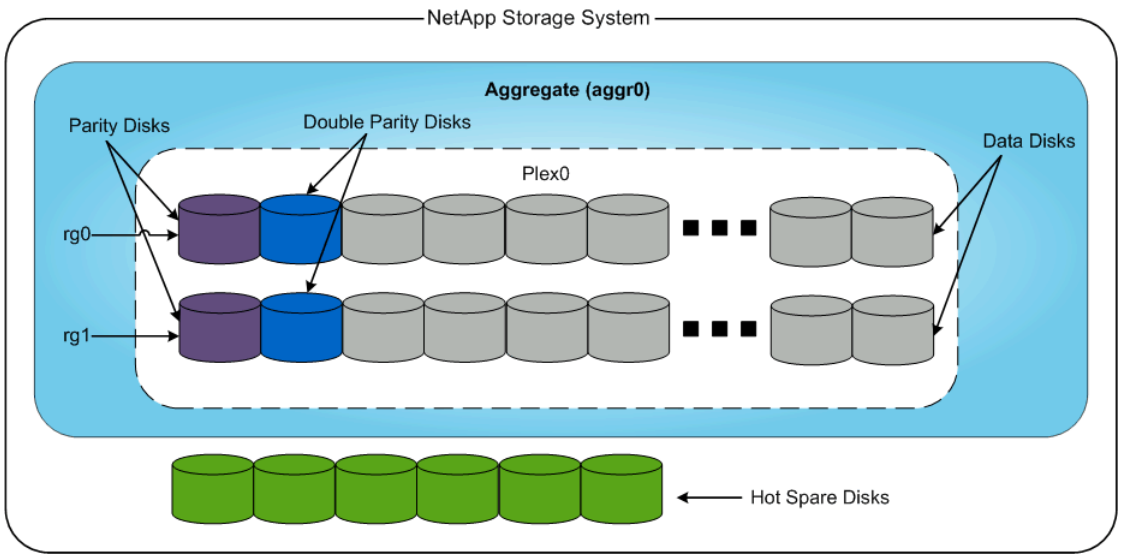

Figure 2) Aggregate on a FAS3050 NetApp storage system.

The Data ONTAP operating system, available on NetApp storage systems, supports a virtual storage layer known as a flexible volume, or FlexVol® volume. A FlexVol volume is created within an aggregate and provides greater performance and manageability than traditional volumes. A FlexVol volume can also grow and shrink as needed and spans all the disk spindles contained in a single aggregate.

A flexible volume can be created by executing the following command on the NetApp storage system:

`vol create` [*VolName*][*AggrName*][*VolSize*]

Where:

- *VolName* identifies the name of the volume to be created
- *AggrName* identifies the name of the aggregate that contains the volume

- *VolSize* identifies the size of volume in KB, MB, or GB

For example, to create a volume named dbstorage_data1 within an aggregate named aggr0 and assign it a size of 2GB, one would execute the following command:

```
vol create dbstorage_data1 aggr0 2G
```

For better performance and manageability it is recommended that transaction logs and tablespace containers reside on separate volumes. For this study, two volumes (`dbstorage_data1` and `dbstorage_data2`) were created for the data containers, and one volume (`dbstorage_log`) was also created for the transaction logs. The database directory was created on a volume named `dbstorage`, as illustrated in Figure 1. IBM JFS2 file systems were created on the data volumes.

5.2 ENABLING JUMBO FRAMES

In addition to having an efficient DBMS and storage system, jumbo frames was enabled on both the data server and the storage system to obtain rapid communication between the data server and the storage system. If the connection topology involves a switch, the switch must also have support for jumbo frames enabled.

At the lowest "physical" layer, electrical signals are exchanged on the network, representing bits and bytes. But just above that layer, the network exchanges frames. An Ethernet frame contains 1,500 bytes of user data, plus its headers and trailer. By contrast, a jumbo frame contains 9,000 bytes of user data, so the percentage of overhead for the headers and trailer is much less, and data transfer rates can be much higher.

On the NetApp storage systems, jumbo frames can be enabled by executing the following command:

```
ifconfig e10 mtusize 9000 up
```

To make the above setting persistent, include the `ifconfig` statement in the `/etc/rc` file on the storage system. The `ifconfig` line in the `/etc/rc` file on the IBM N5500 storage system used in these tests was:

```
ifconfig e5 [hostname] -e5 netmask 255.255.255.0 mtusize 9000 flowcontrol full
```

5.3 MODIFYING VOLUME OPTIONS

There are certain options that can be enabled to achieve good performance from a NetApp storage system when running database workloads. The following options were used in our environment:

- **Disabled automatic Snapshot™ copies**. By default when a volume is created, it has a default Snapshot schedule set for it. Normally a database is backed up based on a user-defined schedule; therefore there is no need for volumes used by databases to use the default Snapshot schedule. Automatic Snapshot copies were disabled on each of the volumes used by the database with the following command:

  `vol options` [*VolName*] **`nosnap on`**

  Where:

  VolName identifies the name of the volume

- **Disabled the read-ahead feature**. Historically, NetApp had recommended disabling aggressive storage readahead for OLTP (Online Transaction Processing) database workloads by setting the Data ONTAP parameter "`minra`" to "`on`". Data ONTAP 6.5.1, however, introduced significant changes to the readahead algorithm, making it more intelligent and efficient. Hence, disabling readahead for database workloads is no longer recommended. Recent experience indicates that in fact, enabling `minra` may lower the overall database performance. As a result, NetApp now recommends that the `minra` setting be left in the default "`off`" state unless explicit guidance to do otherwise is given by the NetApp Global Support Organization.
- **Updated the access time of all files**. If this option is on, it prevents the update of the access time on an inode when a file is read (each file has an inode that stores information about that file). For volumes used by databases, the correct access time for inodes will be managed by the DBMS; therefore this option was enabled by executing the following command on each volume:

vol options [*VolName*] **no_atime_update on**

Where:

VolName identifies the name of the volume

- **Disabled the Snapshot directory display**. By default the Snapshot directory is visible to users at the client mountpoints. If this option is on, the display of the Snapshot directory at client mountpoints is disabled. The Snapshot directory display for each volume was disabled in our environment using the following command:

  vol options [*VolName*] **nosnapdir on**

  Where:

  VolName identifies the name of the volume

- **Set nvfail option on**. If this option is on, the NetApp storage system performs additional status checking at boot time to verify that the NVRAM of the storage system is in a valid state. This option is useful for databases, because if any problems with NVRAM are found, the database instances shut down, and an error message is sent to the console to alert database administrators. The following command was used to set the nvfail option for each volume:

  vol options [*VolName*] **nvfail on**

  Where:

  VolName identifies the name of the volume

- **Set the Snapshot reserve to zero**. When a new volume is created, by default Data ONTAP reserves 20% of space for the Snapshot copies, which can not be used for the data. In order to better utilize the storage space, we opted to set the Snapshot reserve to 0 by executing the following command:

  snap reserve -V [*VolName*]**0**

  Where:

  VolName identifies the name of the volume

To change all of the options mentioned above for a volume named `dbstorage_data2`, execute the following set of commands:

```
vol options dbstorage_data2 nosnap on
vol options dbstorage_data2 minra on
vol options dbstorage_data2 no_atime_update on
vol options dbstorage_data2 nosnapdir on
vol options dbstorage_data2 nvfail on
snap reserve -V dbstorage_data2 0
```

For other configuration and suggestions on the storage system's physical design, refer to the technical report "DB2 9 for UNIX: Integrating with a NetApp Storage System" [2].

6 AIX 5L PERFORMANCE TUNING

There are several operating system parameters that can be tuned to gain better performance from the data server. In our setup, the DB2 9 database runs on AIX 5L 5.3 TL04. As outlined before, IBM JFS2 file systems were created on the data volumes, which allow higher performing file system options such as CIO to be used. In addition to this, buffer cache paging activity on the data server was controlled and AIO tuning was performed.

6.1 CONTROLLING BUFFER-CACHE PAGING ACTIVITY

The behavior of the AIX 5L file buffer cache manager can have a significant impact on performance because excessive paging activity can decrease performance substantially. It can also cause an I/O bottleneck, resulting in lower overall system throughput. On AIX 5L, tuning buffer-cache paging activity must be done carefully and infrequently.

The `lru_file_repage` parameter instructs the system to steal file memory pages only when determining what type of memory to steal. This means that the file system pages are reused in preference to the database pages, resulting in better database performance. This can be enabled by executing the following command:

```
vmo -o lru_file_repage=0
```

For more details on this parameter, see "IBM System p and AIX information Center" [3].

6.2 ASYNCHRONOUS I/O TUNING

An application using synchronous I/O cannot continue until the I/O operation it is waiting on is complete. In contrast, asynchronous I/O (AIO) operations run in the background and do not block user applications. AIO improves performance because I/O operations and application processing can run simultaneously. The actual performance, however, depends on how many server processes that handle I/O requests are running.

For this study, the AIO parameters "`maxservers`" and "`maxreqs`" were updated by executing the following commands:

```
aioo -o maxservers=20

aioo -o maxreqs=32768
```

For more details on AIO performance benefits, see "Database Performance Tuning on AIX" [8].

7 NETWORK TUNING

A dedicated Gigabit Ethernet network that connects the storage system and the data server is recommended to obtain high bandwidth. The Gigabit Ethernet driver can also play an important role in network performance. In our environment, version 5.3.0.40 was used, along with a 1 Gigabit Ethernet network adapter per storage system for the NFS and iSCSI protocols.

As outlined before, jumbo frames were enabled for these tests. For an Ethernet adapter jumbo frames can be enabled by executing the following command on the data server:

```
chdev -l '[AdapterName]' -a jumbo_frames='yes' -a mtu='9000'
```

Where *AdapterName* identifies the name of the Ethernet adapter on which the jumbo frame is enabled

For example, to enable jumbo frames and set mtu size to 9000 for an Ethernet adapter named `en02`, execute the following command:

```
chdev -l 'en02' -a jumbo_frames='yes' -a mtu='9000'
```

8 PROTOCOL-SPECIFIC TUNINGS

8.1 AIX 5L NFS CONFIGURATION

For the NFS protocol, tuning must be performed at the data server and storage system levels.

SERVER TUNING.

The following mount options were used for the data file systems:

```
options=rw,bg,hard,nointr,proto=tcp,vers=3,rsize=32768, wsize=32768,timeo=600
```

Additionally, the parameter `nfs_rfc1323` was enabled, which allows for TCP window sizes greater than 64KB and thus helps minimize the wait for TCP acknowledgments. For more details, see "AIX 5L NFS Client Performance Improvements for Databases on NAS" [7]. The `nfs_rfc1323` parameter can be set by executing the following command on the data server:

```
nfso -o nfs_rfc1323=1
```

STORAGE SYSTEM TUNING.

The following parameter settings were used on the NetApp storage system:

```
nfs.tcp.enable on
nfs.tcp.recvwindowsize 65536
nfs.tcp.xfersize 65536
```

The first option verifies that the storage system will accept TCP connections. The next two parameters set the receive window size and the NFS transfer size to their maximum values. These parameters can be set using the options command, as shown in the following example:

```
options nfs.tcp.enable on
```

8.2 AIX 5L ISCSI CONFIGURATION

The AIX 5L iSCSI driver has a default `queue_depth` of 1, which is too low for database workloads. In this setup, the `queue_depth` was changed to 256 for each hdisk available by executing the following command (for example, for hdisk1):

```
chdev -l hdisk1 -a queue_depth=256
```

The maximum transfer size was also changed to 512KB for each hdisk available by executing the following command:

```
chdev -l hdisk1 -a max_transfer=0x80000
```

In addition, the following network parameters were used:

```
no -p -o tcp_recvspace=262144
no -p -o tcp_sendspace=262144
no -p -o tcp_nagle_limit=0
no -p -o ipsrcrouterecv=1
no -p -o sb_max=1310720
```

For more details on this parameter, see the "IBM System p and AIX information Center" [3].

8.3 AIX 5L FCP CONFIGURATION

The queue depth for each disk was changed to 256 using the command described in section 8, subsection 2. Additionally, `num_cmd_elems`, which is a setting for the maximum number of commands to queue to the FC-AL card adapter, was changed to 2,048 using the following command:

```
chdev -l fcs0 -a num_cmd_elems=2048
```

9 RESULTS

The results of these experiments were determined by the average throughput (transactions attained per minute) attained after a 15-minute warm-up and over a steady state of 20 minutes while running an OLTP workload. A total of 250 users was used so that most of the CPU on the data server was utilized.

Figure 3 shows the relative throughput attained while using the FCP, iSCSI, and NFS protocols. The results show that the throughput achieved for each protocol was within 20%; with the maximum attained from FCP. NFS yielded 7% more throughput than iSCSI.

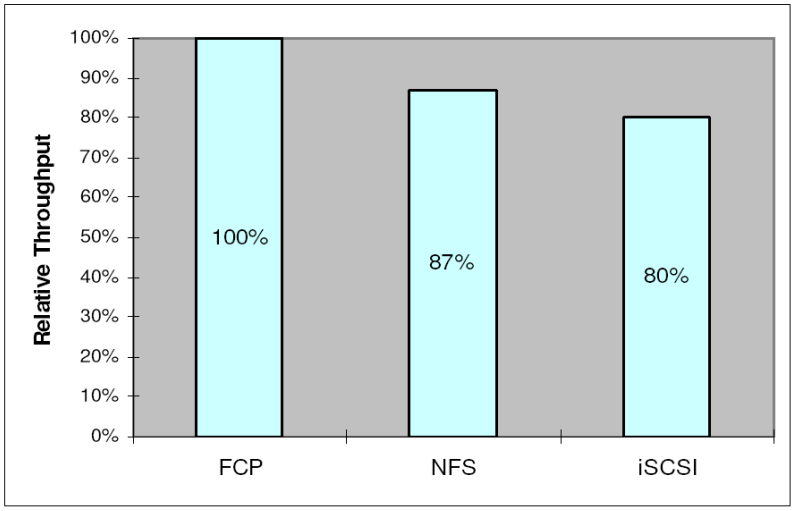

Figure 3) Relative throughput for each protocol.

Figure 4 shows the CPU utilization during the workload runs for each of the protocols. A higher percentage of time is spent in the kernel in the NFS and iSCSI cases. The path length of a transaction is longer in these cases, as compared to FCP, since more kernel components are involved in processing the transaction. An explanation of the extra kernel components that are involved is beyond the scope of this paper.

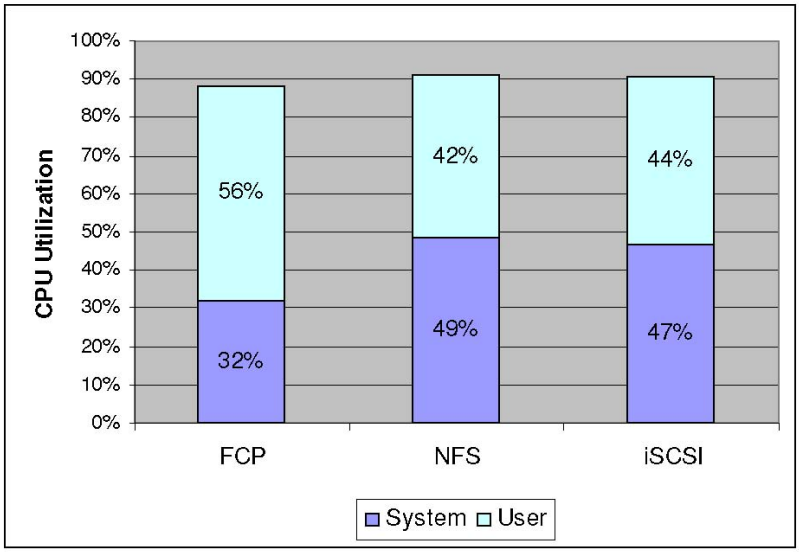

Figure 4) CPU utilization during OLTP workload runs for each protocol.

The scaled relative throughput was also derived and is shown in Figure 5; this was calculated by dividing the actual throughput attained for each of the protocols with the percentage of user plus kernel CPU used by each protocol, and then using the number obtained for FCP as the 100% point. The results echo the results from Figure 3, with FCP bearing the best throughput per CPU percentage. The results also show that FCP has an advantage over NFS and iSCSI when CPU utilization is taken into account.

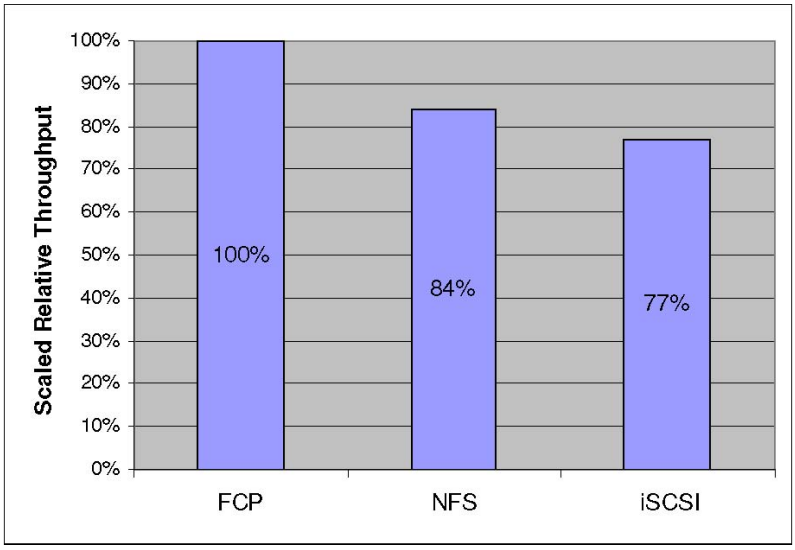

Figure 5) Scaled relative throughput for each protocol.

10 CONCLUSION

This paper demonstrates that with IBM DB2 9 and AIX 5.3 TL-04, using an IBM N5500 system as storage, the throughput attained with NFS is comparable to that of the iSCSI protocol. FCP yielded higher throughput compared to the other two protocols due to less kernel CPU usage. However, an advantage of NFS compared to SAN is accessibility by clients via networks. FCP connectivity currently is a slightly more expensive solution that does not offer this added accessibility.

11 REFERENCES

1) IBM System Storage N Series Implementation of Raid Double Parity for Data Protection: www.redbooks.ibm.com/redpapers/pdfs/redp4169.pdf

2) DB2 9 for UNIX: Integrating with NetApp Storage System: www.netapp.com/library/tr/3531.pdf

3) IBM System p and AIX information Center: publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp

4) Fibre Channel: Overview of the Technology: www.fibrechannel.org/technology/overview.html

5) ISCSI Protocols Concepts and Implementation: www.cisco.com/en/US/netsol/ns340/ns394/ns259/ns261/networking_solutions_white_paper09186a00800a90e4.shtml

6) Improve Database Performance on File System Containers in IBM DB2 UDB V8.2 Using Concurrent I/O on AIX: www128.ibm.com/developerworks/db2/library/techarticle/dm-0408lee/

7) AIX 5L NFS Client Performance Improvements for Databases on NAS: www-03.ibm.com/servers/aix/whitepapers/aix_nfs.pdf

8) Database Performance Tuning on AIX: www.redbooks.ibm.com/redbooks/SG245511/wwhelp/wwhimpl/java/html/wwhelp.htm

9) DB2 UDB Exploitation of NAS Technology: www.redbooks.ibm.com/abstracts/sg246538.html

12 REVISION HISTORY

Date	Author	Comments
May 2007	Jawahar Lal and Roger Sanders	Original draft
May 2009	Esther Smitha	Updated changes to the readahead setting recommendation.