NETAPP TECHNICAL REPORT

Oracle 11*g* Single Instance Database Sequential Workload Performance Comparison with Direct NFS and iSCSI on Windows Server 2003 Enterprise x64 Edition

Performance and Scalability Case Study Using NetApp Storage

Saad Jafri, NetApp
June 2008 | TR-3622

ABSTRACT

This technical report compares the performance of Oracle® Direct NFS (DNFS) protocol with iSCSI using a sequential workload on Oracle 11*g*™ single instance databases running on 64-bit Microsoft® Windows® Server 2003 Enterprise x64 Edition and NetApp® storage. The focus of this paper is technical; the reader should be comfortable with Windows administration, Oracle 11*g* database administration, network connectivity, iSCSI administration, and NetApp storage administration.

TABLE OF CONTENTS

1 INTRODUCTION AND SUMMARY

This technical report discusses the scalability and performance of Oracle Direct NFS (DNFS) under a sequential workload and compares its performance with iSCSI using the same workload.

DNFS is an NFS client developed by Oracle. Instead of relying on the operating system, the Oracle DNFS client generates NFS requests directly from the database engine.

This Oracle client implementation delivers several unique advantages:

- Oracle can use the same NFS client, regardless of the platform it is running on, resulting in more consistent and predictable performance.
- Because the client spans platforms, bugs can be identified and fixed faster.
- This is the first NFS client validated for Oracle on Windows.
- On UNIX® platforms, no mount option tuning is necessary. Existing kernel-based NFS clients require specific mount options that differ for each UNIX flavor.
- DNFS provides native support for easily distributing NFS workloads across up to 16 network links. This has been difficult with traditional NFS clients.

This report details the specific environment, configurations, and performance results of the performance testing comparing DNFS and iSCSI. The primary result of this study indicates that DNFS scales well in terms of network throughput and can generate throughput comparable to iSCSI.

2 HARDWARE AND SOFTWARE ENVIRONMENT

2.1 STORAGE CONFIGURATION

The NetApp storage system configuration is shown in Table 1.

Table 1) NetApp storage system configuration.

Component	Details	
Operating system	Data ONTAP® 7.2.3	
Storage system model	FAS6070	
Disks	Six shelves containing a total of 84 FC disks, each 144GB, 15K RPM	
Connections from storage controllers to disk shelves	Three active FC ports were used to connect to the shelves	
Storage connections to server	DNFS tests used one 1Gbps to four 1Gbps Ethernet connections	iSCSI tests used four 1Gbps Ethernet connections

2.2 SERVER CONFIGURATION

The server configuration is shown in Table 2. An HP DL585 G2 system hosted the Oracle 11*g* application and generated the sequential workload used in the tests. Any supported SMP server platform could be used to generate the workload. Results may vary, depending on the server platform used.

Table 2) Server configuration.

Component	Details
Operating system	Microsoft Windows Server 2003 Enterprise x64 Edition Service Pack 2
System type	HP DL585 G2
Processor	Four dual core 2.4GHz AMD Opteron
Physical RAM	32GB
Swap space	64GB
Database server	Oracle 11.1.0.6.0
iSCSI host attach kit	NetApp iSCSI Windows Host Utilities 4.0 for Native OS and Veritas
iSCSI software initiator	Microsoft Software Initiator 2.05
Network connections to storage	Two PCI-X Intel® Pro/1000 MT quad-port NICs (used two GbE ports on each NIC)

3 TEST DESCRIPTION

The tests conducted were designed to generate a heavy sequential workload, as typically seen in DSS-type database environments. The purpose of these tests was to generate enough sequential workload to saturate the bandwidth of the network links from the server to the storage and demonstrate the network throughput scalability by using DNFS while increasing the number of Gigabit Ethernet (GbE) links from one to four. Adding network links for DNFS was straightforward and easy to implement. For example, no additional NIC bonding software or driver was needed for load balancing. For network I/O load balancing, DNFS automatically manages the I/Os across multiple NICs. The only step required to add a network link was to specify the IP address for the additional network path in the `oranfstab` configuration file used by DNFS.

In addition, the same test was executed with iSCSI using four GbE links to compare the network throughput with DNFS using four GbE links.

The single instance Oracle database created for testing was approximately 750GB in size. The database emulated a wholesale supplier's database in which inventory is spread out across several regional warehouses. It included tables for customers, inventory stock, warehouses, historical information, and indexes. One of the larger tables was the customer table, with 150 million rows. Full table scan test queries were run against this customer table with Oracle's parallel query feature (degree of parallelism).

The primary objective of the tests was to show the performance and scalability of DNFS using NetApp storage as the available network bandwidth on the Windows server is scaled from one to four GbE links. Five different tests were performed, shown in Table 3. The following measurements were taken for each test:

- Average throughput in Mbytes/sec achieved during each test as seen by the Oracle database in the AWR report
- Average host CPU utilization on the Windows server during each test as seen by the Oracle database in the AWR report

Table 3) Test number and associated details for each test executed.

Test #	Test Query	Degree of Parallelism	# of Network Connections	Storage Protocol
1	select count(*) from cust	8	One GbE	DNFS
2	select count(*) from cust	8	Two GbE	DNFS
3	select count(*) from cust	8	Three GbE	DNFS
4	select count(*) from cust	8	Four GbE	DNFS
5	select count(*) from cust	8	Four 1GbE	iSCSI

Each query in the tests shown in Table 3 performs a select operation against a large table called cust, which contains 150 million rows.

4 DNFS TEST RESULTS

Figure 1 shows the network throughput versus host CPU utilization reported by the Oracle database during the execution of the following test query as the number of GbE links scaled from one to four:

select count(*) from cust	Degree of parallelism = 8; cust table contains 150 million rows

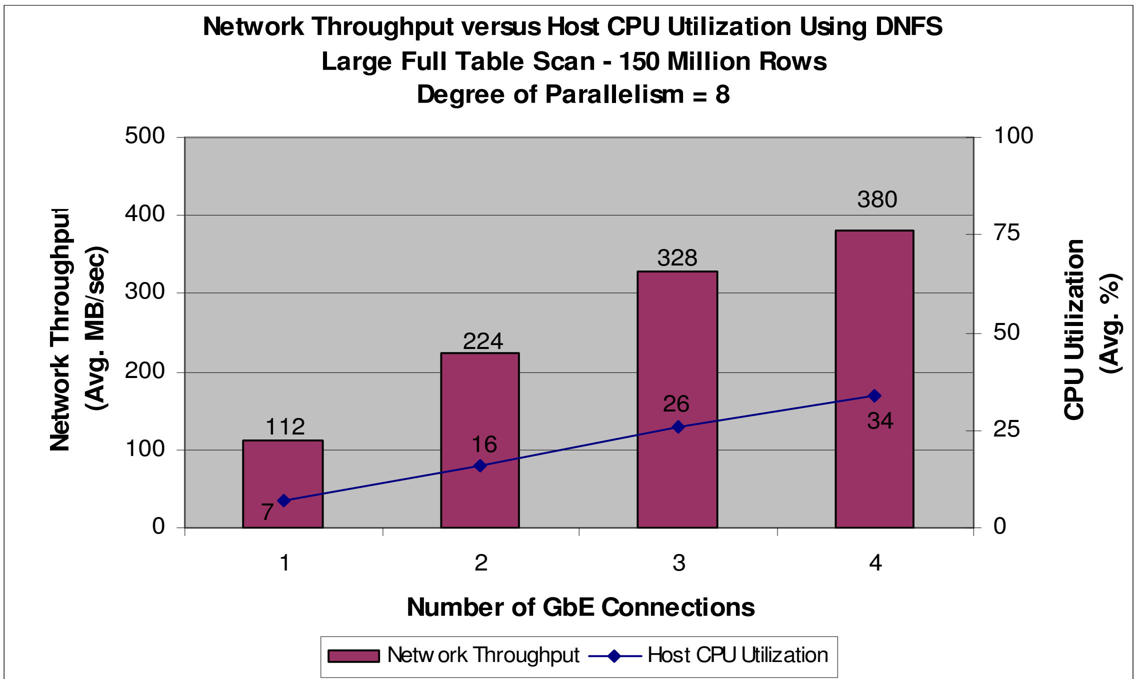

Figure 1) Network throughput versus host CPU utilization using DNFS as the number of GbE links increased from one to four.

Conclusions based on the results shown in Figure 1:

- Going from one to two GbE links yields 100% (112MB/sec to 224MB/sec) improvement in network throughput.
- Going from two to four GbE links yields 70% (224MB/sec to 380MB/sec) improvement in network throughput.
- Host CPU utilization also increases as the number of GbE links scales from one to four.

Figure 2 shows the read latency reported by the Oracle database during the execution of the following test query as the number of GbE links scaled from one to four:

select count(*) from cust	Degree of parallelism = 8; cust table contains 150 million rows

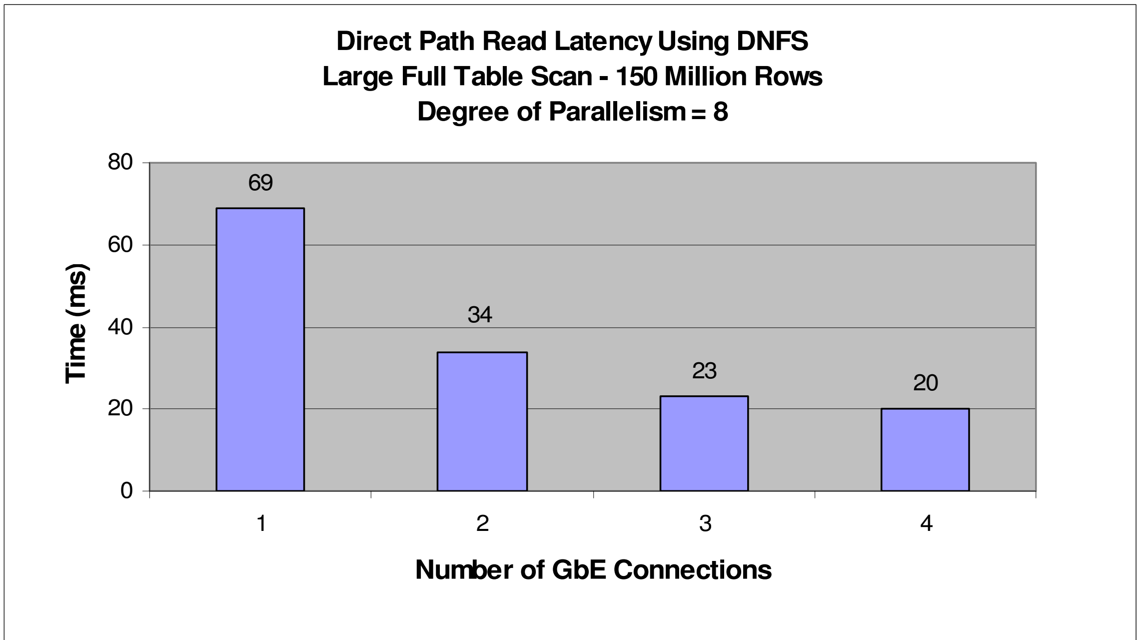

Figure 2) Direct path read latency using DNFS as number of GbE links scales from one to four.

Note: In Figure 2, lower read latency times indicate better performance. As one to four GbE links are added, the read latency decreases correspondingly, resulting in higher overall throughput.

Conclusions based on the results shown in Figure 2:

- Going from one to two GbE links yields 103% (69ms to 34ms) improvement in read latency.
- Going from two to four GbE links yields 70% (34ms to 20ms) improvement in read latency.

In all tests using DNFS, the NetApp storage system easily met the demands of the test queries with resources to spare. As demonstrated by the results in Figure 1, network throughput performance with a sequential workload scales well over DNFS as additional GbE links are added.

5 ISCSI TEST RESULTS

The next set of tests used iSCSI storage interconnects. The iSCSI tests were done using four GbE links from the host to the NetApp storage system. The goal of the test was to compare the performance of DNFS and iSCSI by using comparable bandwidth configurations— four GbE links. Microsoft Windows NTFS file systems were created on the LUNs containing the database and log files. The iSCSI tests were performed using the same test query that was used for the DNFS tests. The amount of data accessed was the same as the DNFS tests; the test query was run against a customer table with 150 million rows.

Figure 3 shows the iSCSI results with four GbE links from the host to the NetApp storage compared to DNFS using four GbE links from the host to the NetApp storage. The following query was run for both tests:

select count(*) from cust	Degree of parallelism = 8; cust table contains 150 million rows

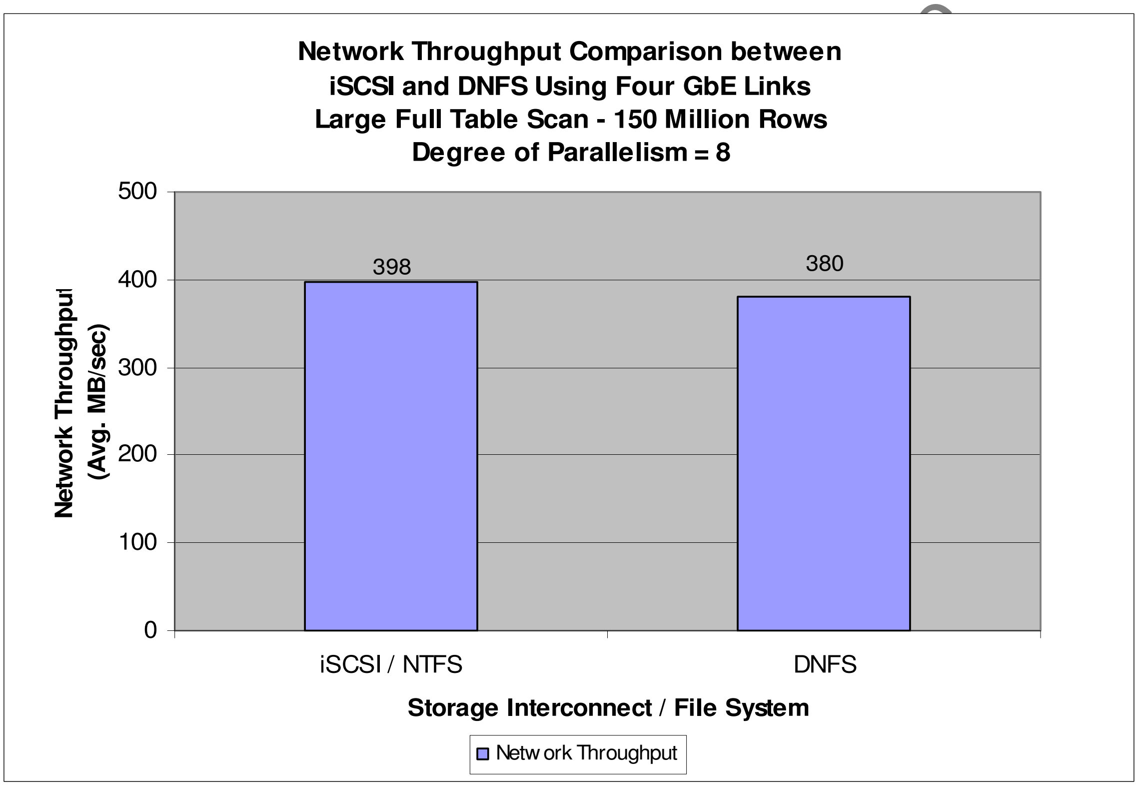

Figure 3) Network throughput comparison for DNFS and iSCSI using four GbE links.

Conclusions based on the results shown in Figure 3:

- Both DNFS and iSCSI effectively used the bandwidth available from all four GbE network links.
- DNFS delivers approximately the same throughput performance compared to iSCSI and is much simpler to configure because no separate driver or MPIO software is needed for DNFS.

Figure 4 compares the read latency reported by the Oracle database during the execution of the test query using four GbE links from host to NetApp storage:

select count(*) from cust	Degree of parallelism = 8; cust table contains 150 million rows

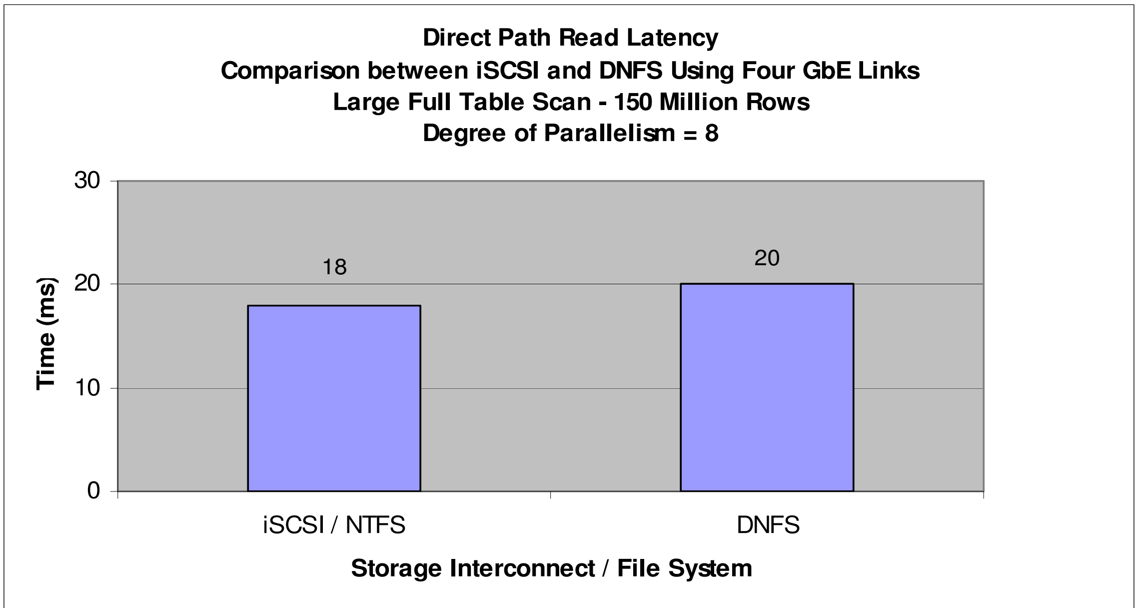

Figure 4) Direct path read latency comparison between DNFS and iSCSI using four GbE links.

Conclusions based on the results in Figure 4 for a sequential workload:

- Using four GbE links from host to NetApp storage, read latency is comparable between DNFS and iSCSI.

In the test using iSCSI, the NetApp storage system easily met the demands of the test query with resources to spare. During the testing in both configurations, DNFS and iSCSI, host CPU utilization was less than 35%. As demonstrated by the results in Figure 3, network throughput performance for a heavy sequential workload is comparable between iSCSI and DNFS. Additionally, DNFS is simpler to set up compared to iSCSI while delivering approximately the same throughput.

6 CONCLUSION

This paper demonstrates that for a sequential workload, DNFS shows good network throughput scaling as the number of GbE links is increased from one to four. Additionally, it shows that four GbE links with DNFS deliver network throughput that is comparable to that of four GbE links with iSCSI. Furthermore, DNFS is simpler to set up than iSCSI because no separate driver and MPIO configuration is needed for DNFS. Future technical reports will look at performance of DNFS in other host operating environment.

APPENDIX A: NETWORK DIAGRAM

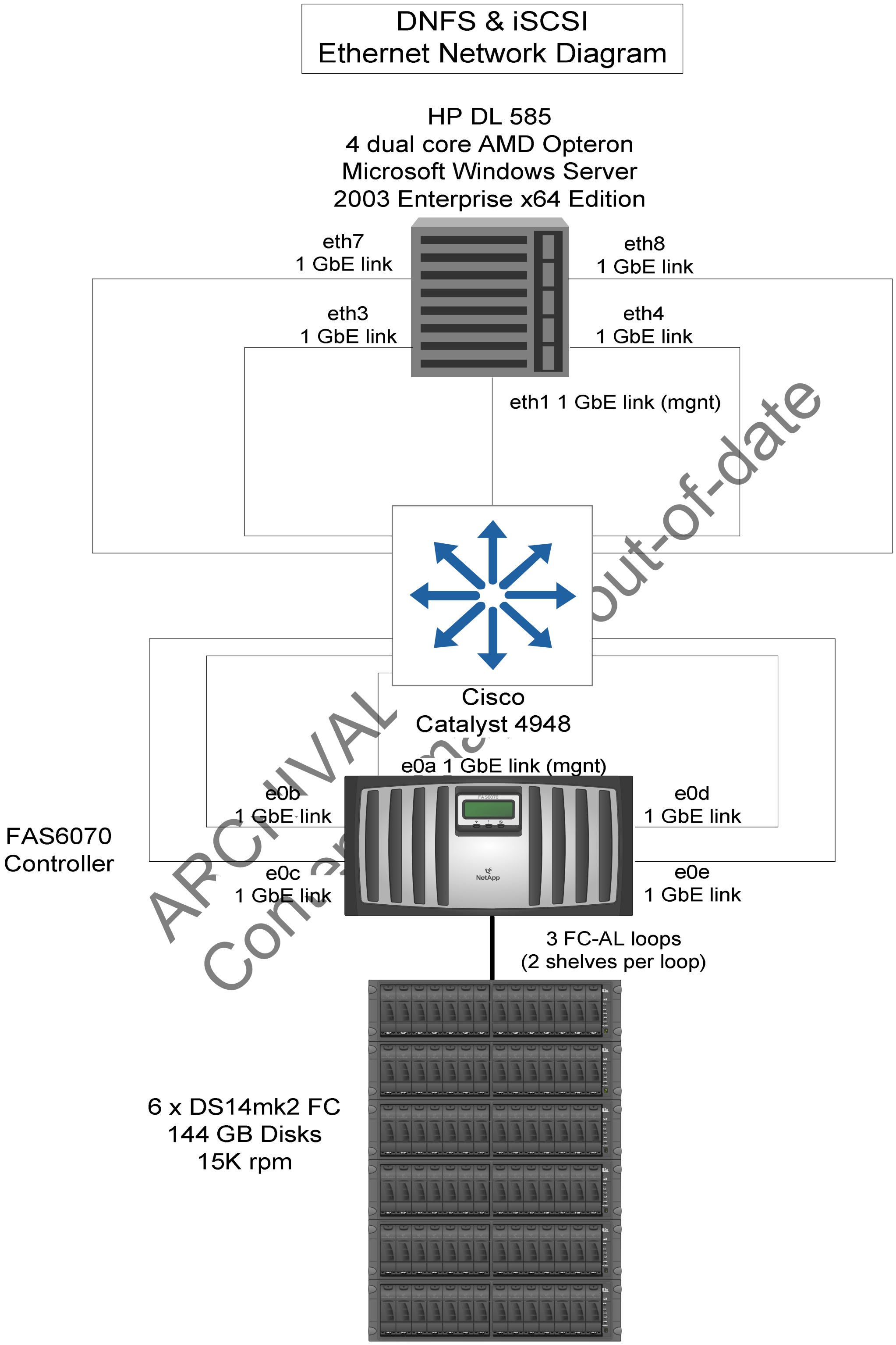

Figure 5) Network for DNFS and iSCSI tests.

APPENDIX B: STORAGE SYSTEM LAYOUT TABLE FOR DNFS TESTS

Table 4) Storage system layout for DNFS tests.

NetApp Storage Controller	Aggregate Name	Volume Name	Size	Description
Controller	Aggr0		3 Disks	
		/vol/vol0	86GB	Root volume
	Aggr1		80 Disks	
		/vol/oradata	2966GB	Oracle data files
		/vol/oralog	1483GB	Oracle log files

All RAID groups were 16 disks in size and configured to use RAID-DP.

APPENDIX C: STORAGE SYSTEM LAYOUT TABLE FOR ISCSI TESTS

Table 5) Storage system layout for iSCSI tests.

NetApp Storage Controller	Aggregate Name	Volume Name	LUN	Size	Description
Controller	Aggr0			3 Disks	
		/vol/vol0		86GB	Root volume
	Aggr1			80 Disks	
		/vol/oradata		2966GB	Oracle data files volume
			/vol/oradata/datalun1	950GB	Oracle data files LUN
		/vol/oralog		1483GB	Oracle log files volume
			/vol/oralog/loglun1	100GB	Oracle log files LUN

All RAID groups were 16 disks in size and configured to use RAID-DP.

APPENDIX D: DNFS CONFIGURATION

Oracle 11*g* software was installed using the Oracle installer on the Windows server. During the installation process, "software only" was selected instead of creating a starter database.

Once the Oracle 11*g* database software was installed, the following steps were taken to enable DNFS:

1. Created a file called `$ORACLE_HOME/dbs/oranfstab` to contain the following information:

```
server:       controller1
path:  192.168.20.20
path:  192.168.21.20
path:  192.168.22.20
path:  192.168.23.20
export: /vol/oradata mount: D:\app\oracle\oradata\data
export: /vol/oralog mount: D:\app\oracle\oradata\log
```

2. Oracle database uses an ODM library, oranfsodm11.dll, to enable Direct NFS. To replace the standard ODM library, oraodm11.dll, with the ODM NFS library, oranfsodm11.dll:
   a. Change directory to `$ORACLE_HOME\bin`.
   b. Shut down Oracle.
   c. Enter the following commands:
      i. `copy oraodm11.dll oraodm11.dll.stub`
      ii. `copy /Y oranfsodm11.dll oraodm11.dll`

To map the DNFS mount points used by Oracle to the NetApp storage volumes:

1. On the NetApp storage controller, an IP address and a unique hostname were assigned to each network interface port. Each IP address was configured on a separate subnet to ensure appropriate network traffic flows through the desired interface.
2. On the Windows server, an IP address was assigned to each of its network interface ports. Each IP address was defined on a separate subnet that matched a corresponding interface on a storage controller. In other words, one server IP address and one storage controller IP address were defined in each subnet.

To separate management traffic from DNFS I/O, dedicated management NICs were used on the server and NetApp storage controllers. Table 6 illustrates IP address configuration on the Windows server and storage controller used for DNFS traffic.

Table 6) IP address and NIC assignements for DNFS tests.

NetApp Storage Controller	Physical Network Interface	IP Address	Host Name	Description
Controller	e0a	10.61.175.208	controller1	For management traffic
	e0b	192.168.20.20	controller1-e0b	For DNFS traffic
	e0c	192.168.21.20	controller1-e0c	For DNFS traffic
	e0d	192.168.22.20	controller1-e0d	For DNFS traffic
	e0f	192.168.23.20	controller1-e0e	For DNFS traffic
Windows Host				
Server	eth1	10.61.175.136	hpdl585	For management traffic
	eth3	192.168.20.35	hpdl585-eth3	For DNFS traffic
	eth4	192.168.21.35	hpdl585-eth4	For DNFS traffic
	eth7	192.168.22.35	hpdl585-eth7	For DNFS traffic
	eth8	192.168.23.35	hpdl585-eth8	For DNFS traffic

Directories were created to hold Oracle data on the NetApp storage controller, as shown in Table 7.

Table 7) Directory layout under /vol/oradata and /vol/oralog on NetApp storage controller.

NetApp Storage Controller	Volume Name	Directory Name	Description
Controller	/vol/oradata		Oracle data volume
		data	Contains Oracle data files
	/vol/oralog		Oracle log volume
		log	Contains Oracle log files

Note: On Windows platforms, two optional parameters can be specified in the `oranfstab` file:

- uid: UNIX User ID to be used by Direct NFS
- gid: UNIX Group ID to be used by Direct NFS

The Direct NFS client uses the uid or gid value to access all NFS servers listed in `oranfstab`. Direct NFS ignores uid and gid values of 0. If neither uid nor gid is specified, then a default of uid:65534, gid:65534 is used by the Direct NFS client. The default value often corresponds to user:nobody and group:nogroup on the NFS server. For these tests, the defaults of uid:65534 and gid:65534 were used.

Note: The exported path from the NFS server must be accessible for read, write, and execute operations by the user with the uid, gid specified in `oranfstab`. If neither uid nor gid is listed, then the exported path must be accessible by the user with uid:65534, gid:65534.

Because of the access requirement noted above for effective uid/gid, appropriate read/write persmissions can be granted on the directory location serving DNFS by using a third- party NFS client such as Microsoft Services for UNIX running on the Windows system or NFS mounting the directory location on another UNIX system authenticated by either NIS or a local password file.

For the testing outlined in this report, granting the appropriate read/write permissions was done by NFS mounting the directory locations via another UNIX system authenticated by a local password file. The /vol/oradata and /vol/oralog volumes on the NetApp controller were NFS mounted on another UNIX system, and data and log directories were created under the respective volumes. The following commands were executed on the directories to grant the necessary ownership and read/write permissions:

- `chown oracle:ORA_DBA data`
- `chmod 755 data`
- `chown oracle:ORA_DBA log`
- `chmod 755 log`

APPENDIX E: ISCSI CONFIGURATION

Physical network connections were made as per the network diagram shown in Appendix A, Figure 5.

The following steps were taken to configure iSCSI connectivity between the server and the NetApp storage controllers:

1. Created LUNs on the NetApp storage as per the data shown in Appendix C, Table 5.
2. Installed Microsoft Software Initiator 2.05 with Microsoft MPIO multipathing support for iSCSI on the server.
3. Installed the NetApp iSCSI Windows Host Utilities 4.0 for Native OS and Veritas™ package on the server per the NetApp Setup Guide.
4. Using the Microsoft iSCSI Initiator GUI, added target portals for each of the four servers' IP addresses. For each target portal IP, the source IP on the NetApp controller was selected to be on the same subnet as the target portal IP.
5. Target was configured as follows:
   - To use multipathing
   - To be persistent across reboots
   - To establish one session with four different connections (source and target portal IPs being on the same subnet) using round-robin load balancing policy
6. In the Windows Disk Management applet on the server, the LUNs discovered were formatted with NTFS and a unique drive letter was assigned.

Table 8 shows the mapping of LUNs to mount points used by the Oracle database.

Table 8) Mapping of LUNs to mount points.

Storage Controller	LUN	Mountpoint Drive Letter / Directory	Description
Controller	/vol/oradata/datalun1	E:\oradata	Oracle data files
	/vol/oralog/loglun1	F:\oralog	Oracle log files

APPENDIX F: ORACLE DATABASE CONFIGURATION

The single instance Oracle database created for testing was approximately 750 gigabytes in size. The database emulated a wholesale supplier's database in which inventory is spread out across several regional warehouses. It included tables for customers, inventory stock, warehouses, historical information, and indexes. As part of the database creation process, a customer table was created with 150 million rows, against which the full table scan test queries were executed to generate sequential workload.

Table 9 shows the initialization parameters in the `init.ora` file that were used for all of the sequential workload test queries:

Table 9) Oracle initialization parameters.

Parameter Name	Begin Value
_in_memory_undo	FALSE
_undo_autotune	FALSE
compatible	10.2.0.0.0
control_files	**For DNFS tests:** D:\APP\ORACLE\ORADATADNFS\DATA\CONTROL_001, D:\APP\ORACLE\ORADATADNFS\DATA\CONTROL_002 **For iSCSI tests:** E:\ORADATA\CONTROL_001, E:\ORADATA\CONTROL_002
cursor_space_for_time	TRUE
db_16k_cache_size	10737418240
db_2k_cache_size	268435456
db_4k_cache_size	268435456
db_block_size	8192
db_cache_size	10737418240
db_files	129
db_name	tpcc
dml_locks	3000
log_buffer	10485760
parallel_max_servers	100
plsql_optimize_level	2
processes	1200
recovery_parallelism	40
sessions	2300
shared_pool_size	4294967296
statistics_level	typical
transactions	2600
undo_management	AUTO
undo_retention	2
undo_tablespace	undo_1

ACKNOWLEDGEMENTS

Special thanks to the following for their contributions:

NetApp

Lee Dorrier, Steve Daniel

NetApp™

www.netapp.com