



PERFORMANCE REPORT: ORACLE10G™ RAC ON LINUX®

A Performance Comparison Study of Oracle® Real Application Clusters Against
FCP, NFS, and iSCSI

John Elliott | Network Appliance | January 2006 | TR 3423

TABLE OF CONTENTS

Performance Report: Oracle10g RAC On Linux.....	1
Introduction	3
Section 1: Executive Summary	4
Figure 1) Overall Database Throughput Comparison For All Test Configurations	6
Section 2: Performance Tuning	6
Figure 2) Cumulative Impact Of Each Incremental Change	8
Section 3: Test Configurations	8
Test Configuration 1: Oracle10g RAC With ASM Using iSCSI LUNs And Software Initiators.....	9
Figure 3) Network And Hardware Configuration Of RAC Environment With ASM, iSCSI LUNs And iSCSI Software Initiators	10
Test Configuration 2: Oracle10g RAC With ASM Using NFS Files.....	10
Table 2) CPU Utilization Comparison Between ASM With NFS And ASM With iSCSI Software Initiators	11
Figure 4) Network And Hardware Configuration Of ASM Environment Using NFA Files.....	12
Test Configuration 3: Oracle10g RAC With NFS As A Standalone File system.....	12
Figure 5) Network And Hardware Configuration Of NFS Environment	14
Test Configuration 4: Oracle10g RAC With ASM Using QLogic QLA4010 iSCSI HBAs	14
Figure 6) Network And Hardware Configuration Of Oracle10g RAC/ASM/iSCSI with QLA4010 iSCSI HBAs	16
Test Configuration 5: Oracle10g RAC With ASM Using QLogic QLA2342 Fibre Channel Adapters	16
Figure 7) Network And Hardware Configuration Of Oracle10g RAC and ASM With QLA2342 Fibre Channel HBAs.....	18
Test Configuration 6: Oracle10g RAC With OCFS And QLogic QLA2342 Fibre Channel HBAs	18
Figure 8) Network And Hardware Configuration Of Oracle10g RAC/OCFS With QLA2342 Fibre Channel HBAs	20
Section 4: Results Comparison And Summary	20
Figure 9) Overall Database Throughput For All Test Configurations	21
Figure 10: CPU Utilization Comparison Across Configurations.....	23
Conclusion.....	23
Appendix A – Lowering The SGA Attach Address.....	25
Appendix B – How To Enable Asynchronous I/O In Oracle	25
Appendix C – Linux Kernel Parameter Settings.....	26
Appendix D – Queue Depth Limitations For Software iSCSI Initiators	27
Appendix E – NFS Mount Options	27
Appendix F – Relevant Oracle Parameter Settings.....	27
Appendix G – Comment On Jumbo Frames	27

Introduction

Recently a series of tests were conducted by Network Appliance, Inc., to demonstrate the high level of performance that can be obtained from an Oracle10g RAC database using Network Appliance™ storage systems. Six different 3-node RAC configurations were tested using NFS, iSCSI and Fibre Channel Protocol (FCP), running on NetApp FAS980 storage systems. All tests were performed using RedHat Enterprise Linux Advanced Server 3.0 in support of the Oracle low-cost storage initiative. Test results clearly support Network Appliance storage systems as high performing, low-cost, feature-rich storage for Oracle10g RAC databases. Before continuing with the details of this study we should first consider the advantages of the Oracle10g RAC database platform.

In terms of data management, Oracle10g Real Application Clusters provide the scalability, availability and data protection required in today's 24x7 world of enterprise computing. Key features of Oracle10g RAC include:

- Support of rolling software upgrades and patch applications to greatly reduce the downtime once required for system maintenance
- The ability to recover from server hardware failure, basically removing the database host as a single point of failure
- A rich set of services and service control tools, greatly enhancing the management and manageability of the database environment
- Improved computing capacity management by allowing physical nodes to be added to the cluster without incurring expensive downtime
- The ability to use smaller, less expensive servers for databases with heavy I/O requirements, that without the scalability of RAC would require much larger servers with more physical memory and higher CPU count

Another important feature of Oracle10g which is not unique to the RAC environment is its "Automatic Storage Management" (ASM) feature. ASM was designed to provide I/O performance comparable to that of raw disk storage with much of the convenience and ease of management associated with a file system. With ASM, storage LUNs are grouped into ASM disk groups. These disk groups act as cluster-enabled storage containers for Oracle data files, control files, log files, and spfiles. LUNs can be added to or removed from disk groups on the fly with no overall impact on data availability, greatly enhancing the ability of the administrator to manage storage. In support of this "on the fly" storage provisioning, ASM provides for both automatic and deferred data rebalancing once the capacity of a disk group has been altered. Rebalancing basically results in the redistribution of data across the disks in a disk group for optimization of storage performance.

Oracle Real Application Clusters require shared storage which is accessible to all nodes in the cluster. This requirement can be satisfied in a number of ways, including but not limited to Oracle ASM, Oracle Cluster File System (OCFS), raw disks and NFS. Network Appliance storage systems provide feature rich storage for all four of these configurations as follows:

- NFS mounted volumes for storage of datafiles, control files, voting file, cluster registry file, log files, pfiles, spfiles and Oracle Home
- iSCSI LUNs for raw disk implementations, ASM disk groups, OCFS partitions and any other cluster-enabled file system
- FCP LUNs for raw disk implementations, ASM disk groups, OCFS partitions and other cluster-enabled file systems
- Files in NFS mounted volumes for use as ASM disks

In these environments, NetApp storage systems provide superior cost-effective data protection, backup and recovery, availability, and administration through Network Appliance tools and features which include the following:

- Fast reliable backups using Snapshot™, SnapVault® and NearStore® technologies
- Cloning and database refresh tools
- SnapManager® for Oracle backup and recovery
- Disk redundancy through RAID and RAID-DP™
- Storage system redundancy through the use of cluster technology
- Extensive array of disaster recovery tools

Section 1: Executive Summary

The Oracle10g RAC database used for these tests included a total of three Dell PowerEdge 2650 servers, each configured with dual 3.2 GHz processors and 4GB of RAM running Red Hat Enterprise Linux 3.0 Update 4. We chose the Linux operating system because it so closely aligns with the universal goals of most enterprise IT departments, to achieve high performance and high availability at the lowest cost of ownership while ensuring data integrity. We tested the following Oracle10g RAC storage configurations:

- ASM using iSCSI LUNs with software initiators
- ASM using iSCSI LUNs with QLogic host bus adapters (HBAs)
- ASM using FCP LUNs
- ASM using NFS files as ASM disks
- NFS mounted volumes
- OCFS using FCP LUNs

We designed the test environments to stress the database server software and hardware with the goal being to totally utilize all database server CPU resources for each round of tests. To this end, we eliminated any other resource bottlenecks in other areas including physical memory, storage and network bandwidth by connecting the three clustered systems using a Gigabit Ethernet switch and providing multiple NetApp FAS980 storage systems on which to store the database files.

Workload Description

The database used for testing can best be described as OLTP in nature with a physical size of approximately 350 gigabytes. During the testing, we used a set of scripts and executables to generate an OLTP type load consisting of a steady stream of small, random read and write operations (approximately 57% reads and 43% writes) against the test database. This workload was designed to emulate the real life activities of a wholesale supplier order processing system in which inventory is spread across several regional warehouses. Within that framework, a single order consisted of multiple database transactions with orders averaging 10 items each. In terms of actual database transactions, each item ordered resulted in all of the following database transactions:

- 1 row selection with data retrieval
- 1 row selection with data retrieval and update
- 1 row insertion

The database utilized both primary and secondary keys for data access. In terms of measured database throughput, the metric of interest was defined as the number of orders processed per minute. Throughout this document, this measurement will be referred to as “order entry transactions per minute” or simply OETs.

In a typical non-RAC database, requests for data blocks are made to the Oracle system global area (SGA). Those blocks not residing in the SGA are read into the SGA from disk. In an Oracle Real Application Clusters database, the SGAs of all instances in the RAC are shared. Simply put, if one instance needs a block that already exists in the SGA of another instance in the same RAC database, that block is copied to the SGA of the requesting instance via the node interconnect, with locks being managed globally. While this is a very simplistic explanation of how Oracle RAC cache fusion works, it does give an idea of the added complexity of RAC I/O and also facilitates an understanding of the importance of using a high speed private network for the node interconnect. Additional benefits of cache fusion are:

- I/O between SGAs in the RAC is much faster than disk I/O.
- A considerably larger memory space is available to each instance.

The point to be made by this is that in the RAC environment, not only do we have I/O between the database hosts and storage systems, but we also have I/O between instances. It should also be noted that the interconnect between RAC nodes uses the UDP network protocol instead of TCP. For these reasons, all servers in our test configurations used Gigabit Ethernet for the node interconnect and tuning was performed to optimize UDP performance.

Finally, each individual test cycle consisted of a 12 minute interval including a 6-minute rampup, a 6 minute measurement cycle and a 20 second rampdown. With each test configuration, we ran a sufficient number of consecutive test runs to achieve a repeatable level of throughput.

Results Summary

Though not the best performer in terms of throughput, the configuration using NFS as a standalone file system proved to be very competitive in comparison to the higher performing FCP and hardware iSCSI configurations. NFS came in 12.4% slower than FCP with OCFS, 4.4% slower than FCP with ASM and only 2% slower than hardware iSCSI with ASM. That level of performance, along with its low cost and ease of setup and administration, makes it the obvious best choice for most Oracle10g RAC environments. It must be recognized, however, that different IT organizations have different needs, often based on a complex set of criteria. As demonstrated by these tests, Network Appliance has several high-performance, feature rich options for Oracle10g RAC data storage, giving the enterprise CIO enough choices to effectively deploy NetApp storage systems in a configuration that is consistent with the goals of the organization, while still benefiting from the value-add for which Network Appliance has become known. See Figure 1 for a summary of test results.

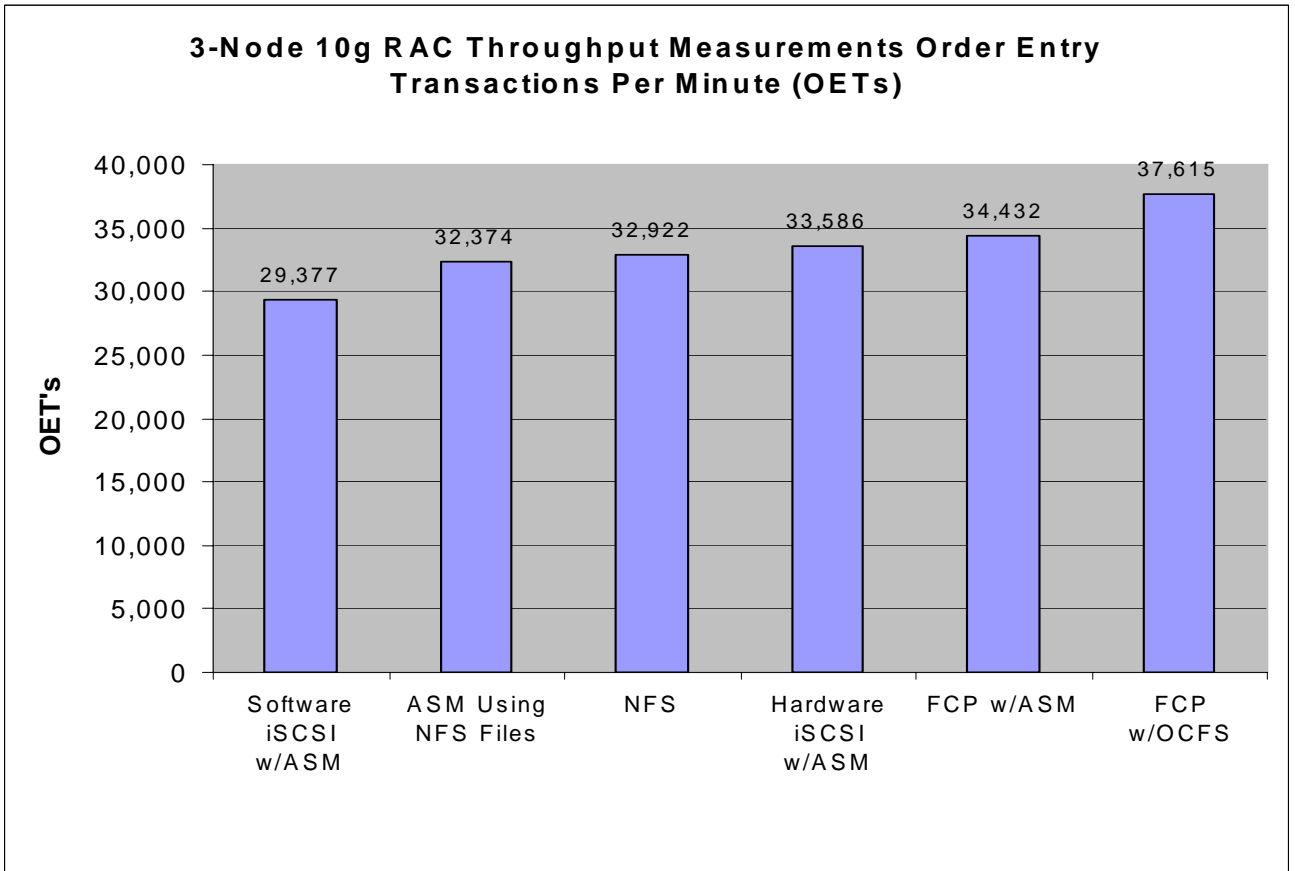


Figure 1) Overall Database Throughput Comparison For All Test Configurations

Section 2: Performance Tuning

Step 1: Establish A Baseline

In order to ensure uniformity across test environments we began by creating a base configuration to serve as a starting point for each environment to be tested. We then conducted a series of preliminary tests to improve the performance of the base configuration by modifying a number of Oracle Database and Linux kernel parameters from their default values. We began by reviewing a number of publicly available NetApp best practices papers as well as general Oracle configuration guidelines. We then conducted a series of tuning iterations in which specific configuration parameters were modified and the resulting performance improvement measured. As a result of this tuning effort, we saw performance improve approximately 78 percent, going from 16,484 OETs using default configuration parameters to 29,377 OETs using our tuned configuration. The remainder of this section provides the details of the tuning operations we performed.

Increase Size of Oracle SGA

Our first step was to create a test database using the Oracle DBCA (Database Configuration Assistant) with default settings for all Oracle initialization parameters. We chose the Oracle10g RAC environment using Oracle Automatic Storage Management with iSCSI software initiators for this testing due to the high level of interest expressed by a large number of partners. For the first test iteration we increased the size of the Oracle SGA from the default value of 1152MB to 2512MB by first recompiling the Oracle executable to use a lower SGA attach address and then

increasing the Oracle “sga_max_size” and the “sga_target” parameters from 1152MB to 2512MB. (The procedure for recompiling the Oracle executable is detailed in Appendix A.) This resulted in an increase in throughput of about 24% compared to the default configuration. (Sizing the SGA to 2512MB yielded a comparable performance gain in all of the remaining configurations tested.) It should be noted that the Linux shared memory kernel parameters had to be set proportionately higher to enable the larger SGA. Those parameters are listed in Appendix C. After increasing the size of the SGA, we observed host CPU utilization in the 60-70% range with no indication of physical memory being a bottleneck. Additionally, we observed no bottlenecks on the FAS980 storage servers used to store the Oracle Database.

Additional Oracle Performance Tuning Tasks

Efforts to increase host side CPU utilization by increasing load along with additional Oracle tuning yielded very little improvement in the overall test results. However, we observed a considerable imbalance in CPU utilization across the three cluster nodes resulting in increases in one node’s CPU idle time with simultaneous decreases in CPU idle time on other nodes. Additionally, we observed that in comparison to other Oracle processes, the LMS (lock manager server process) was using a large amount of CPU time. This was no surprise considering the fact that the LMS process is the key component of the global cache service. The LMS process is responsible for maintaining global cache coherency and moving blocks between instances for cache fusion requests. With this in mind and knowing the impact of global cache performance on RAC database throughput, we increased the scheduling priority of LMS by using the UNIX® “renice” utility to increase the “nice” value of each of the 2 LMS processes running on each of the 3 RAC nodes from 0 to -20. The change resulted in an increase in the scheduling priority of the LMS processes. Adding this change to the previous configuration resulted in an increase in throughput of approximately 32 percent over the previous configuration. At this point database performance statistics looked healthy and host CPU utilization was in the 90-95% range.

Additional observations revealed that Oracle was generating what appeared to be idle parallel execution processes. These processes have names similar to “ora_p001_inst4” and “ora_p002_inst4” with “inst4” being the name of the instance. This behavior is controlled by the Oracle parameters “parallel_max_servers” and “parallel_min_servers”. The default setting is influenced by CPU count and for this database was set to “80” when the database was created. The parallel_min_servers parameter had the default setting of “0”. Setting parallel_max_servers to “0” provided a small amount of improvement as did increasing the “db_writer_processes” parameter from “1” to “2” and decreasing the “parallel_threads_per_cpu” parameter from “2” to “1”. Adding all three changes to the previous configuration resulted in an incremental performance improvement of about 5%.

Finally, we generated another 4% improvement by increasing the db_cache_size parameter from “0” to “2256M”. The default setting of “0” enables the Oracle10g automatic shared memory management (ASMM) process to automatically adjust the size of the default buffer pool on the fly, based on input from the Oracle10g Memory Advisor. ASMM has the ability to manage the large_pool_size, shared_pool_size and java_pool_size in the same manner, based on the sga_target parameter setting. Allowing ASMM to manage the db_cache_size resulted in a buffer cache size of 1968M. Setting db_cache_size to “2256M” forced a minimum size of 2256MB and prevented ASMM from decreasing the size in favor of increasing any of the other three managed values. This change yielded similar improvements in all the remaining configurations tested.

Table 1 summarizes the tuning iterations performed along with the impact of each on database throughput.

	ORACLE PARAMETER	DEFAULT VALUE	NEW SETTING	% INCREASE IN THROUGHPUT

	ORACLE PARAMETER	DEFAULT VALUE	NEW SETTING	% INCREASE IN THROUGHPUT
Tuning Iteration 1	sga_max_size	1152MB	2512MB	24%
	sga_target	1152MB	2512MB	
Tuning Iteration 2	Linux "nice" value of ora_lms processes	0	-20	32%
Tuning Iteration 3	parallel_max_servers	80	0	5%
	db_writer_processes	1	2	
	parallel_threads_per_cpu	2	1	
Tuning Iteration 4	db_cache_size	0	2256MB	4%

Table 1) Tuning Details With Performance Impact

Figure 2 below provides a graphical representation of this data.

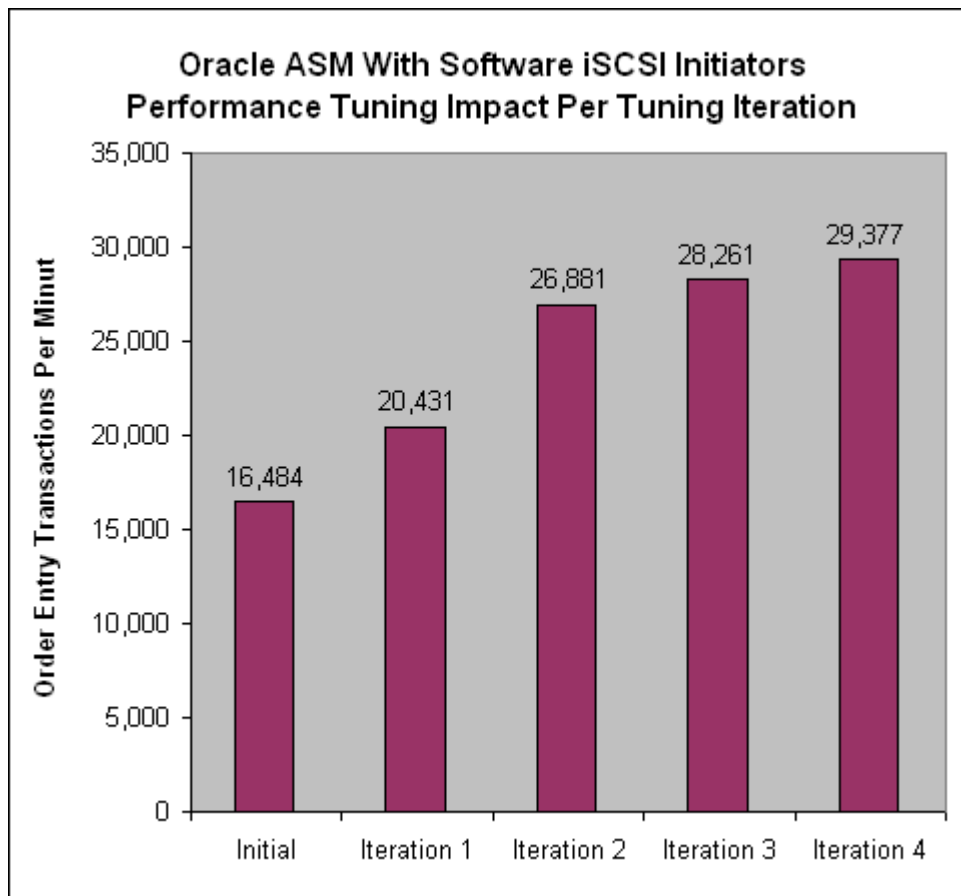


Figure 2) Cumulative Impact Of Each Incremental Change

Section 3: Test Configurations

This section provides the details for each of the six configurations we tested. All test configurations used an identical set of Linux kernel options designed to maximize database performance. These kernel parameters are defined and described in Appendix C of this report. Also, please note that the baseline database created in Section 2, "Performance Tuning," was the starting point for each configuration tested.

Test Configuration 1: Oracle10g RAC With ASM Using iSCSI LUNs And Software Initiators

For this round of tests the default storage system setting of "128" for the "iscsi.iswt.max_ios_per_session" parameter proved to be sufficient to ensure no bottlenecks at the LUN level. We also used asynchronous I/O (AIO). AIO is always preferred when it is available. Though not isolated in this exercise as a performance tuning modification, the database performed consistently better with AIO in the iSCSI environment than without it. In order to enable AIO, we recompiled the Oracle executable as described in Appendix B. As previously stated, this configuration was used for determining the starting point for tuning the remaining test configurations. For a detailed description of tuning activities and test results please refer back to "Section 2: Performance Tuning". The storage system and server environments as well as the specific database parameters used for this configuration are listed below.

Storage system Environment:

- (3) Network Appliance FAS980 Storage systems
- (1) 2TB volume made up of 48 x 72GB disks on each storage system for LUNs
- (4) 60GB LUNs on each storage system for ASM disk groups
- (2) additional 150MB LUNs for Oracle voting disk and OCR disk
- Data ONTAP® 6.5.3P4
- (1) Gigabit Ethernet interface per database node, jumbo frames
- iSCSI licensed and enabled
- Storage system options
 - iscsi.iswt.max_ios_per_session=128
 - iscsi.iswt.tcp_window_size=262800
 - iscsi.enable=on

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Linux iSCSI software initiator version 3.6.2-7 (iscsi-initiator-utils-3.6.2-7)
- oracleasm-lib-2.0.0-1
- Linux kernel parameter settings listed in Appendix C

Oracle Parameter Settings

See Appendix F for a list of pertinent non-default settings. Tuning details and results for this environment have already been provided above in "Section 2: Performance Tuning".

Network Configuration

For a network and hardware diagram of this configuration see Figure 3 below.

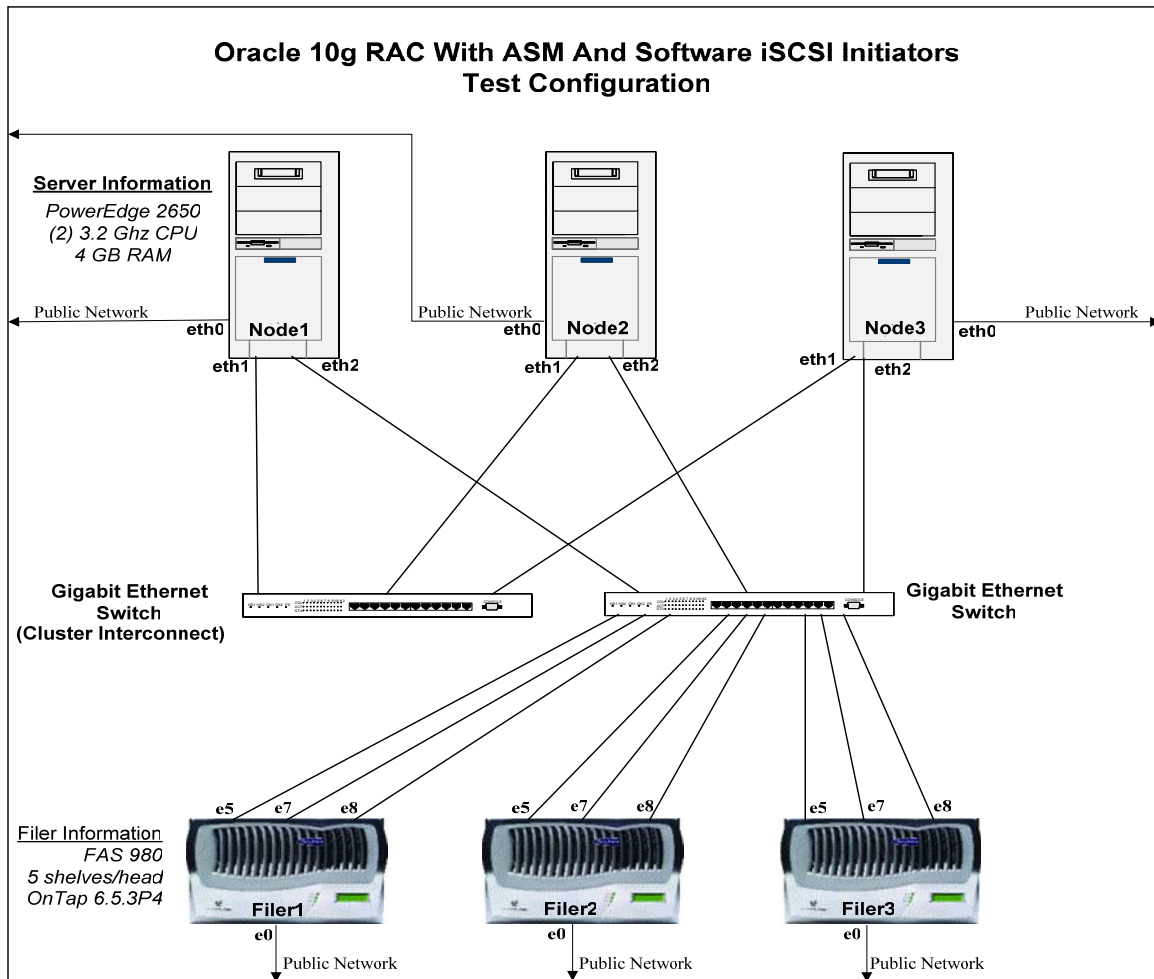


Figure 3) Network And Hardware Configuration Of RAC Environment With ASM, iSCSI LUNs And iSCSI Software Initiators

Test Configuration 2: Oracle10g RAC With ASM Using NFS Files

This section details the configuration used to test a 3-node Oracle10g RAC database using ASM and NFS files, describes the tuning procedure and presents the results. Since asynchronous I/O is not available for NFS on RedHat Enterprise Linux Advanced Server 3 (RHEL 3), multiple db writer processes were absolute necessities for good performance. After a few test iterations we determined that setting the `db_writer_processes` parameter to "8" yielded the highest throughput. Oracle and NetApp best practices require the use of `directio` for RAC databases running on NFS. As a result, we set the parameter `filesystemio_options` to "DIRECTIO". As experienced previously, modifying the `db_file_multiblock_read_count` setting from the default value of "16" gave no improvement. The end result of all tuning iterations in this environment yielded a maximum throughput of 32374 Order Entry Transactions per minute, about 10% higher than the configuration using ASM with iSCSI software initiators. Even with the benefit of asynchronous I/O, the iSCSI software initiator configuration did not perform as well as the ASM environment using

NFS files. vmstat data taken from both environments indicated lower CPU utilization by system processes in the ASM/NFS environment, making more resources available to Oracle user processes, explaining most of the difference in throughput. See Table 2 for more details.

Environment	CPU Utilization by User Processes	CPU Utilization by System Processes
Oracle ASM w/NFS Files	74%	23%
Oracle ASM w/Software iSCSI Initiators	68%	28%

Table 2) CPU Utilization Comparison Between ASM With NFS And ASM With iSCSI Software Initiators

Details of the configuration, both hardware and software, are listed in the sections that follow.

Storage system Environment:

- (3) Network Appliance FAS980 Storage systems
- (1) 2TB volume made up of 48 x 72GB disks on each storage system for LUNs
- (4) 60GB files in each volume for ASM disk groups
- (2) additional 150MB files for Oracle voting file and OCR file
- Data ONTAP 6.5.3P4
- (1) Gigabit Ethernet interface per database node, jumbo frames
- NFS licensed and enabled
- Storage system options
 - nfs.tcp.enable = on
 - nfs.tcp.xfersize = 32768

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Linux kernel parameter settings listed in Appendix C
- NFS mount options used for datafiles:
 - hard,nointr,rsize=32768,wsiz=32768,tcp,actimeo=0,timeo=600
- NFS mount options used for OCR file and Voting file:
 - hard,nointr,rsize=32768,wsiz=32768,tcp,noac,timeo=600

Additional information on NFS mount options can be found in Appendix E.

Oracle Parameter Settings

See Appendix F for a list of non-default settings.

Network Configuration

For a network and hardware diagram of this configuration see Figure 4 below.

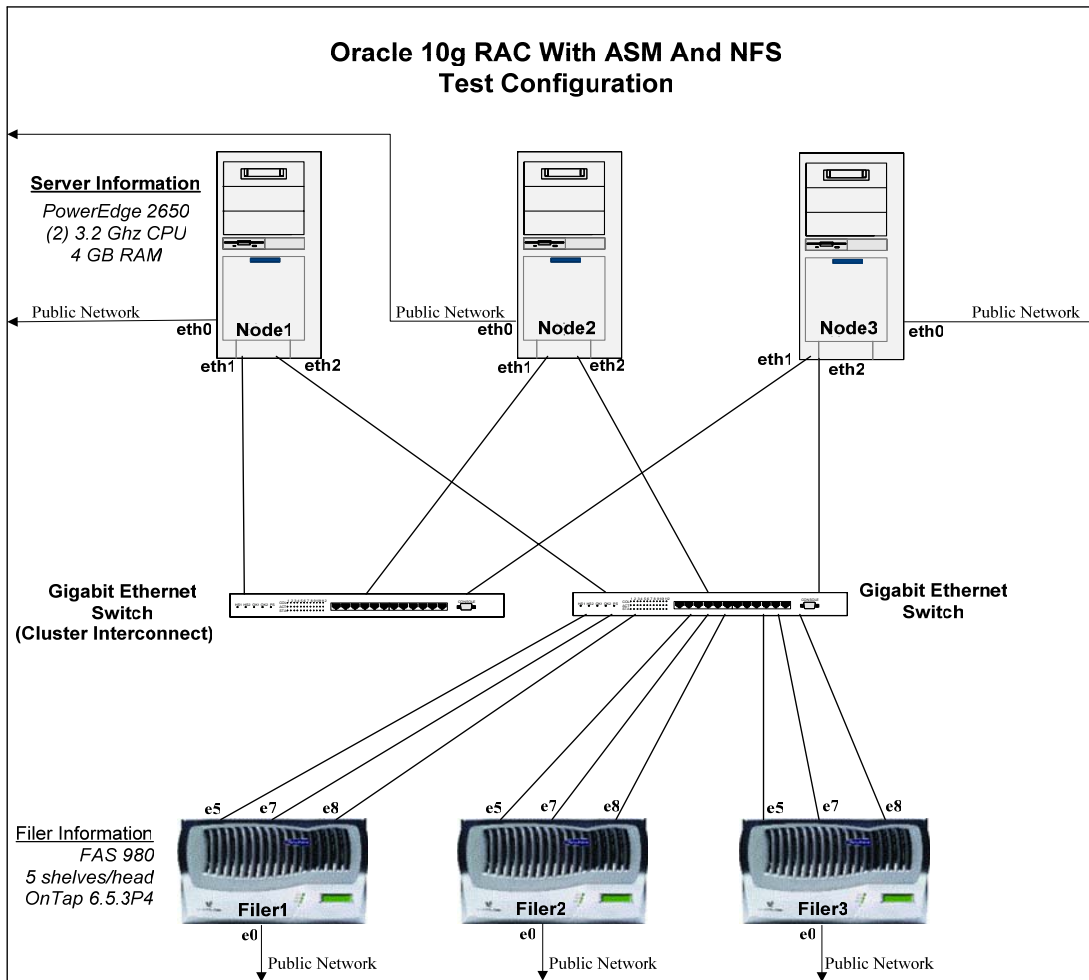


Figure 4) Network And Hardware Configuration Of ASM Environment Using NFS Files

Test Configuration 3: Oracle10g RAC With NFS As A Standalone File system

This section details the configuration used to test a 3-node Oracle10g RAC database using NFS as a standalone file system, describes the tuning procedure and presents the results. Since asynchronous I/O is not available for NFS on RedHat Enterprise Linux Advanced Server 3 (RHEL 3), multiple db writer processes are absolute necessities for good performance. After a few test iterations we determined that setting the "db_writer_processes" parameter to "4" yielded the highest throughput. Oracle and NetApp best practices require the use of directio for RAC databases running on NFS, thus the setting of filesystemio_options=DIRECTIO. As experienced previously, modifying the db_file_multiblock_read_count setting from the default value of "16" gave no improvement. The end result of all tuning iterations in this environment was a maximum throughput of 32922 Order Entry Transactions per minute, about 1.7% higher than the configuration using ASM with NFS files. Host vmstat output from the NFS environment revealed an average of about 94% CPU utilization, about 2% lower than the ASM environment with NFS files. Attempts to drive CPU utilization higher failed. Such a small difference between the 2 configurations in terms of throughput and host CPU utilization makes it very difficult to provide an explicit explanation of the differences. An exhaustive examination of OS and Oracle data yielded no acceptable explanation; however, it can be reasoned that the differences can best be explained as being the result of the increased overhead (processes and memory) created by the

ASM instance in the previous environment. Details of the configuration, both hardware and software, are listed in the sections that follow.

Storage system Environment:

- (2) Network Appliance FAS980 Storage systems – Neither storage system CPU nor storage system memory ever appeared to be a bottleneck in this environment. Due to the perceived potential for higher throughput of those environments using ASM with raw LUNs, 3 heads were used for the other tests.
- (1) 2.6TB volume made up of 64 x 72GB disks on each storage system for datafiles, control files, log files, spfile, voting file and OCR file
- Data ONTAP 6.5.3P4
- (1) Gigabit Ethernet interface per database node, jumbo frames
- NFS licensed and enabled
- Storage system options
 - `nfs.tcp.enable = on`
 - `nfs.tcp.xfersize = 32768`

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Linux kernel parameter settings listed in Appendix C
- NFS mount options used for datafiles:
`hard,nointr,rsize=32768,wsiz=32768,tcp,actimeo=0,timeo=600`
- NFS mount options used for OCR file and Voting file:
`hard,nointr,rsize=32768,wsiz=32768,tcp,noac,timeo=600`

Additional information on NFS mount options can be found in Appendix E.

Oracle Parameter Settings

See Appendix F for a list of pertinent non-default settings.

Network Configuration

For a network and hardware diagram of this configuration see Figure 5 below.

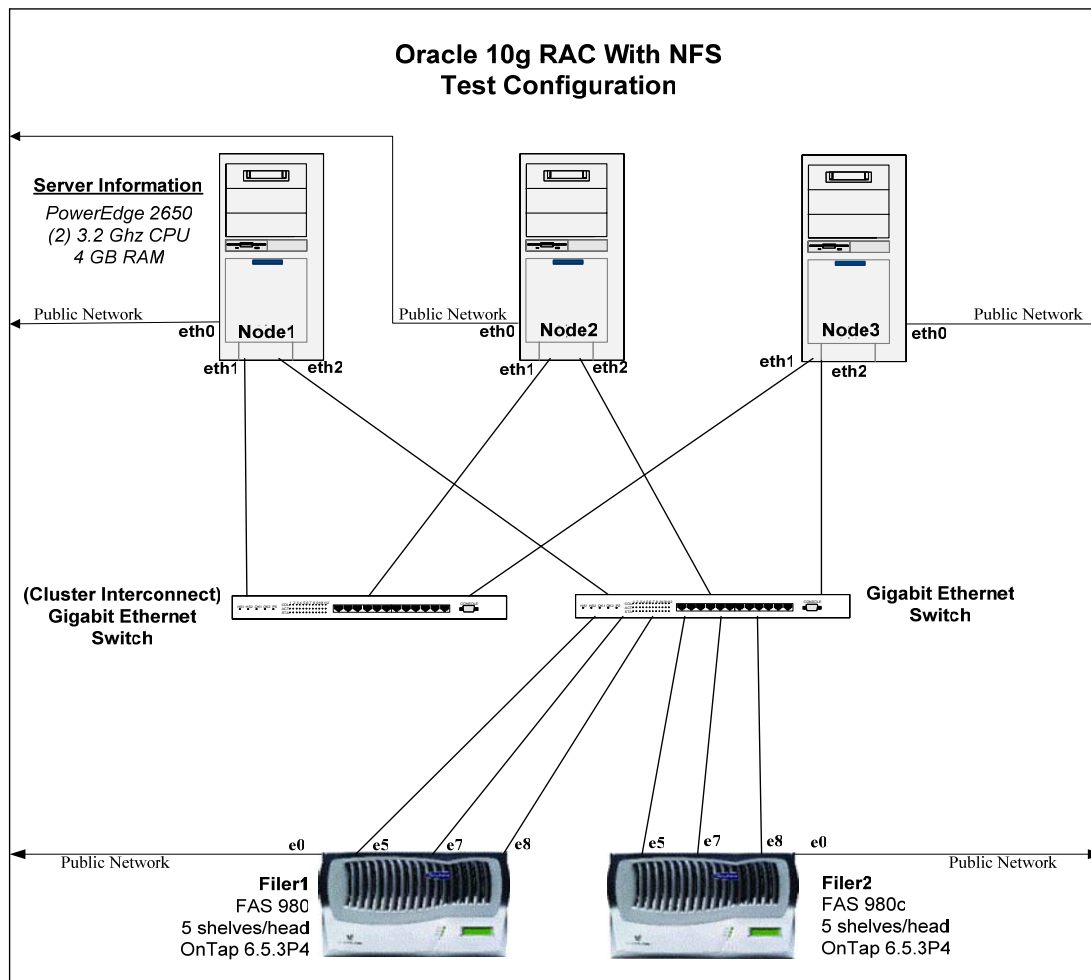


Figure 5) Network And Hardware Configuration Of NFS Environment

Test Configuration 4: Oracle10g RAC With ASM Using QLogic QLA4010 iSCSI HBAs

This section details the configuration used to test a 3-node Oracle10g RAC database using ASM with iSCSI LUNs and QLogic iSCSI HBAs, describes the tuning procedure and presents the results. After a few test iterations we determined that setting the “db_writer_processes” parameter to “2” yielded the highest throughput. As previously experienced, modifying the “db_file_multiblock_read_count” setting from the default value of “16” gave no improvement. The end result of all tuning iterations in this environment yielded a maximum throughput of 33586 Order Entry Transactions per minute, about 14% higher than with ASM and software initiators. Details of the configuration, both hardware and software, are listed in the sections that follow.

Storage system Environment

- (3) Network Appliance FAS980 Storage systems
- (1) 2TB volume made up of 48 x 72GB disks on each storage system for LUNs
- (3) 75GB LUNs on each storage system for ASM disk groups
- (2) additional 100MB LUNs for Oracle voting disk and OCR disk
- Data ONTAP 6.5.3P4
- (1) Gigabit Ethernet interface per database node, jumbo frames
- iSCSI licensed and enabled
- Storage system options

- iscsi.iswt.max_ios_per_session=128
- iscsi.iswt.tcp_window_size=262800
- iscsi.enable=on

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - (1) QLogic QLA4010 iSCSI Host Bus Adapter
 - QLA4010 Firmware v03.00.00.04
 - QLA4010 Driver V3.22
 - execution_throttle=128 (queue depth setting for HBA – set using SANSurfer utility)
 - ql4xmaxqdepth=255
 - max_scsi_luns=128 set in /etc/modules.conf file
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Linux kernel parameter settings listed in Appendix C
- Oracle ASM userspace library, driver support files and kernel driver
 - oracleasm-lib-1.0.0-1
 - oracleasm-support-1.0.3-1
 - oracleasm-2.4.21-EL-smp-1.0.3-1

Oracle Parameter Settings

See Appendix F for a list of non-default settings.

Network Configuration

For network and hardware diagram of this configuration see Figure 6 below.

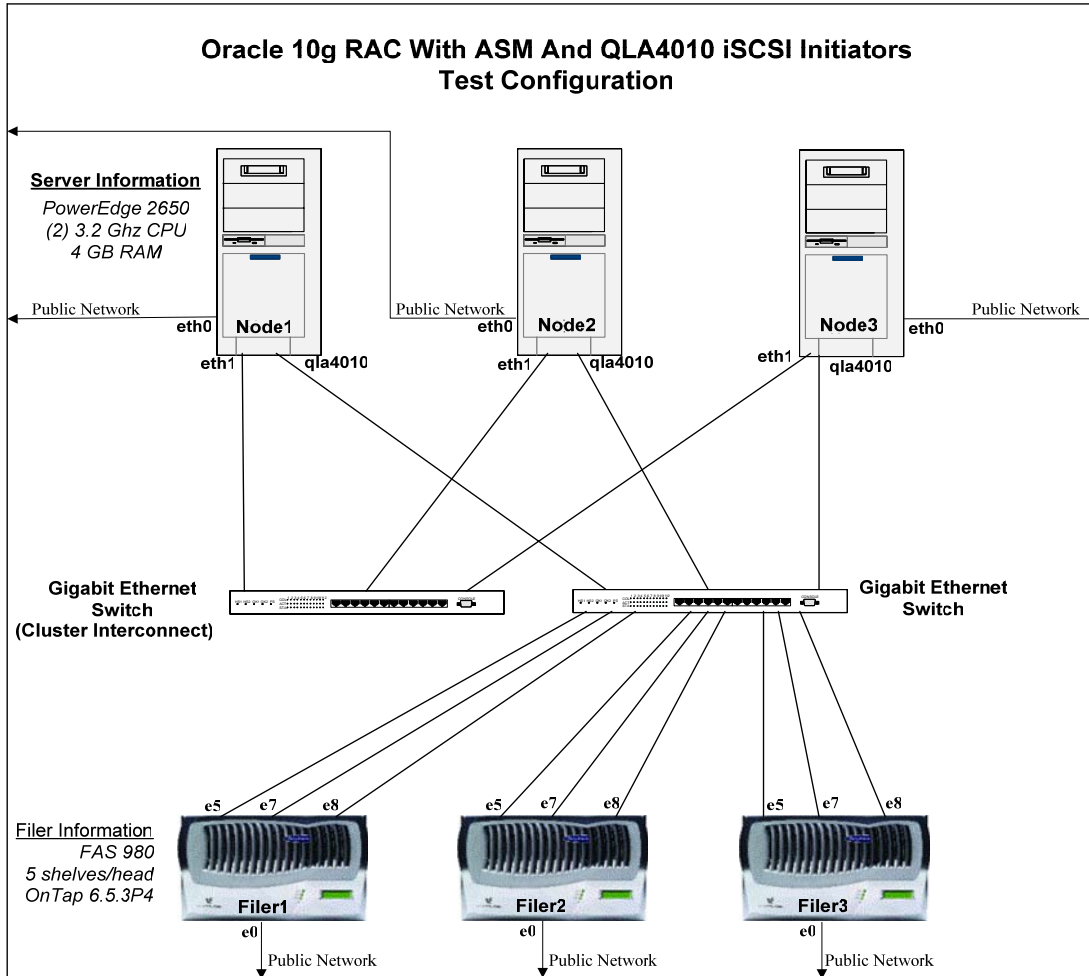


Figure 6) Network And Hardware Configuration Of Oracle10g RAC/ASM/iSCSI With QLA4010 iSCSI HBAs

Test Configuration 5: Oracle10g RAC With ASM Using QLogic QLA2342 Fibre Channel Adapters

This section details the configuration used to test a 3-node Oracle10g RAC database using ASM with FCP LUNS and QLogic FCP Adapters, describes the tuning procedure and presents the results. After a few test iterations we determined that setting the “db_writer_processes” parameter to “2” yielded the highest throughput. As previously experienced, modifying the “db_file_multiblock_read_count” setting from the default value of “16” resulted in no increase in overall performance. The end result of all tuning iterations in this environment yielded a maximum throughput of 34432 Order Entry Transactions per minute, about 2.5% higher than with ASM and iSCSI HBAs. An improvement over iSCSI was expected. Details of the configuration, both hardware and software, are listed in the sections that follow.

Storage system Environment

- (3) Network Appliance FAS980 Storage systems
- (1) 2TB volume made up of 48 x 72GB disks on each storage system for LUNs
- (3) 75GB LUNs on each storage system for ASM disk groups
- (2) additional 100MB LUNs for Oracle voting disk and OCR disk

- Data ONTAP 6.5.3P4
- (1) QLA2342 Fibre Channel Target adapter
 - QLA2342 Firmware v03.03.06
 - execution throttle=128 (queue depth setting for HBA – set using SANSurfer)
 - speed 2 Gb/sec
- FCP licensed and enabled
- Storage system options
 - fcp.enable=on

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - Linux kernel parameter settings listed in Appendix C
 - (1) QLogic QLA2342 Fibre Channel Host Bus Adapter
 - QLA2342 Firmware v03.03.06
 - QLA2342 Driver V7.03.00
 - execution throttle=128 (queue depth setting for HBA – set using SANSurfer)
 - speed 2 Gb/sec
 - ql2xmaxqdepth=256 set in /etc/modules.conf file
 - max_scsi_luns=128 set in /etc/modules.conf file
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Oracle ASM userspace library, driver support files and kernel driver
 - oracleasm-lib-1.0.0-1
 - oracleasm-support-1.0.3-1
 - oracleasm-2.4.21-EL-smp-1.0.3-1

Oracle Parameter Settings

See Appendix F for a list of non-default settings.

Network Configuration

For a graphical representation of this configuration see Figure 7 below.

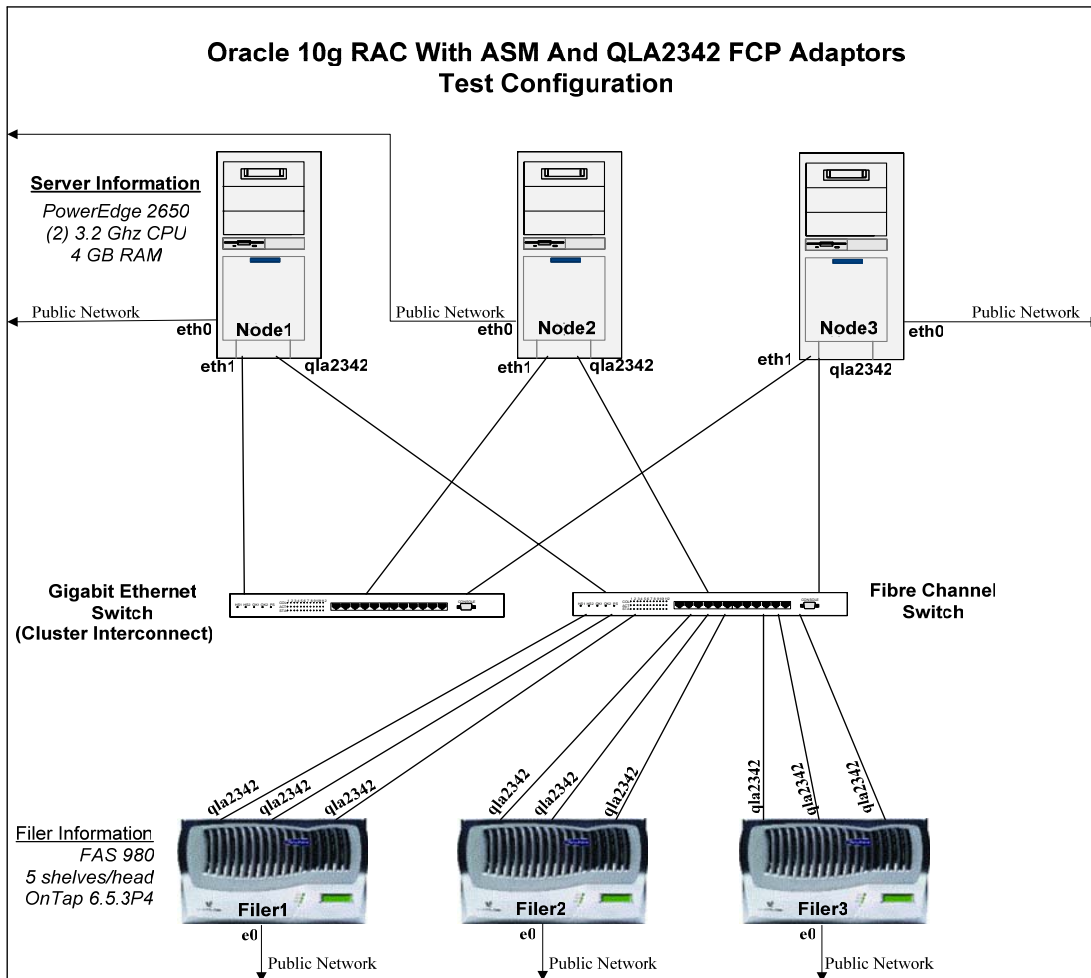


Figure 7) Network And Hardware Configuration Of Oracle10g RAC/ASM With QLA2342 Fibre Channel HBAs

Test Configuration 6: Oracle10g RAC With OCFS And QLogic QLA2342 Fibre Channel Adapters

This section details the configuration used to test a 3-node Oracle10g RAC database using OCFS (Oracle Cluster File system) with FCP LUNS and QLogic FCP Adapters, describes the tuning procedure and presents the results. After a few test iterations we determined that setting the “db_writer_processes” parameter to “2” yielded the highest throughput. As experienced previously, modifying the “db_file_multiblock_read_count” setting from the default value of “16” gave no improvement. The end result of all tuning iterations in this environment yielded a maximum throughput of 37615 Order Entry Transactions per minute, about 9.2% higher than with ASM and FCP. Details of the configuration, both hardware and software, are listed in the sections that follow.

Storage system Environment

- (3) Network Appliance FAS980 Storage systems
- (1) 2TB volume made up of 48 x 72GB disks on each storage system for LUNs

- (1) 250GB LUNs on each storage system for OCFS file systems – to be used for data files
- (1) 80GB LUNs on each storage system for OCFS file systems – to be used for log files
- Data ONTAP 6.5.3P4
- (1) QLogic QLA2342 Fibre Channel Target Adapter
 - QLA2342 Firmware v03.03.06
 - execution throttle=128 (queue depth setting for HBA, set using SANSurfer)
 - speed: 2 Gb/sec
- FCP licensed and enabled
- Storage system options
 - fcp.enable=on

Server Environment:

- (3) Dell PowerEdge 2650 Servers E/W:
 - (2) 3.2 GHz CPU
 - 4GB RAM
 - (1) QLogic QLA2342 Fibre Channel Host Bus Adapter
 - QLA2342 Firmware v03.03.06
 - QLA2342 Driver V7.03.00
 - execution throttle=128 (queue depth setting for HBA, set using SANSurfer)
 - speed: 2 Gb/sec
 - ql2xmaxqdepth=256 set in /etc/modules.conf file
 - max_scsi_luns=128 set in /etc/modules.conf file
 - GbE for cluster interconnect (Jumbo Frames)
 - RHEL 3.0 Update 4
 - Oracle 10.1.0.4.0 Enterprise Edition
- Linux kernel parameter settings listed in Appendix C
- Oracle tools package, support package and kernel driver
 - ocfs-tools-1.0.10-1
 - ocfs-support-1.0.10-1
 - ocfs-2.4.21-EL-smp-1.0.14-1

Oracle Parameter Settings

See Appendix F for a list of non-default settings.

Network Configuration

For a network and hardware configuration diagram see Figure 8 below.

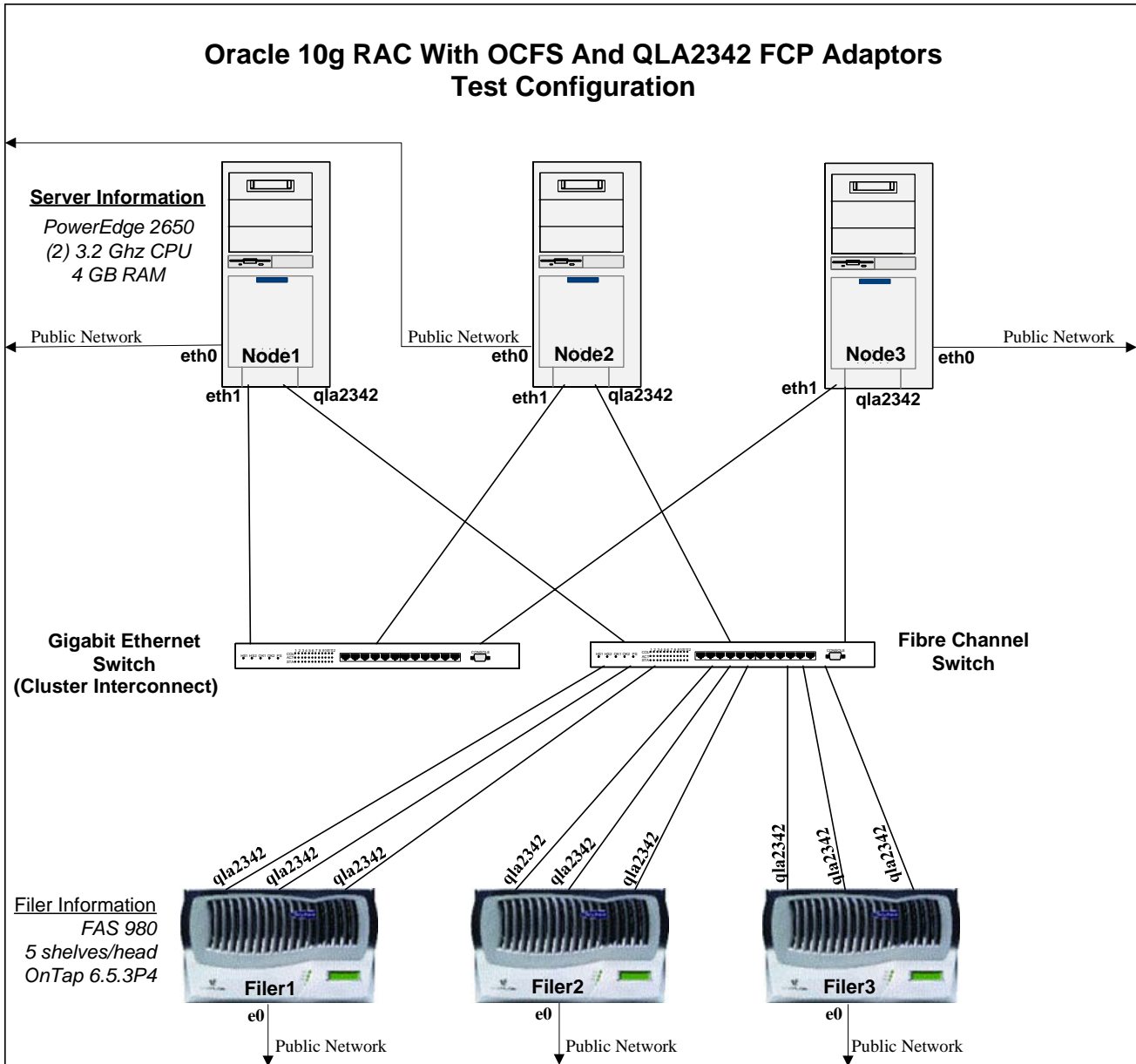


Figure 8) Network And Hardware Configuration Of Oracle10g RAC/OCFS With QLA2342 Fibre Channel HBAs

SECTION 4: RESULTS COMPARISON AND SUMMARY

This final section summarizes and compares key performance test results from each of the six different Oracle10g RAC test configurations described below. It's worth noting that during testing we set up each test configuration in order to stress and fully utilize the database server CPU resources in a 3-node Oracle10g RAC. In doing so, every effort was made to avoid other potential bottlenecks, including disk drives, storage system processing capability, storage system memory and network bandwidth. Below is the list of tested configuration:

- Oracle10g RAC with ASM using iSCSI LUNs and software initiators
- Oracle10g RAC with ASM using NFS mounted files

- Oracle10g RAC with NFS as a standalone file system
- Oracle10g RAC with ASM using iSCSI LUNs and QLogic HBAs
- Oracle10g RAC with ASM using FCP LUNs
- Oracle10g RAC with OCFS using FCP LUNs

Test results are summarized in figure 9. In terms of raw throughput, FCP is clearly the best performer. OCFS was definitely a better performer than ASM. The Fibre Channel solution is a bit costly, including the cost of FCP adapter cards and Fibre Channel switches. For environments requiring FCP performance, however, it is available and performs very well.

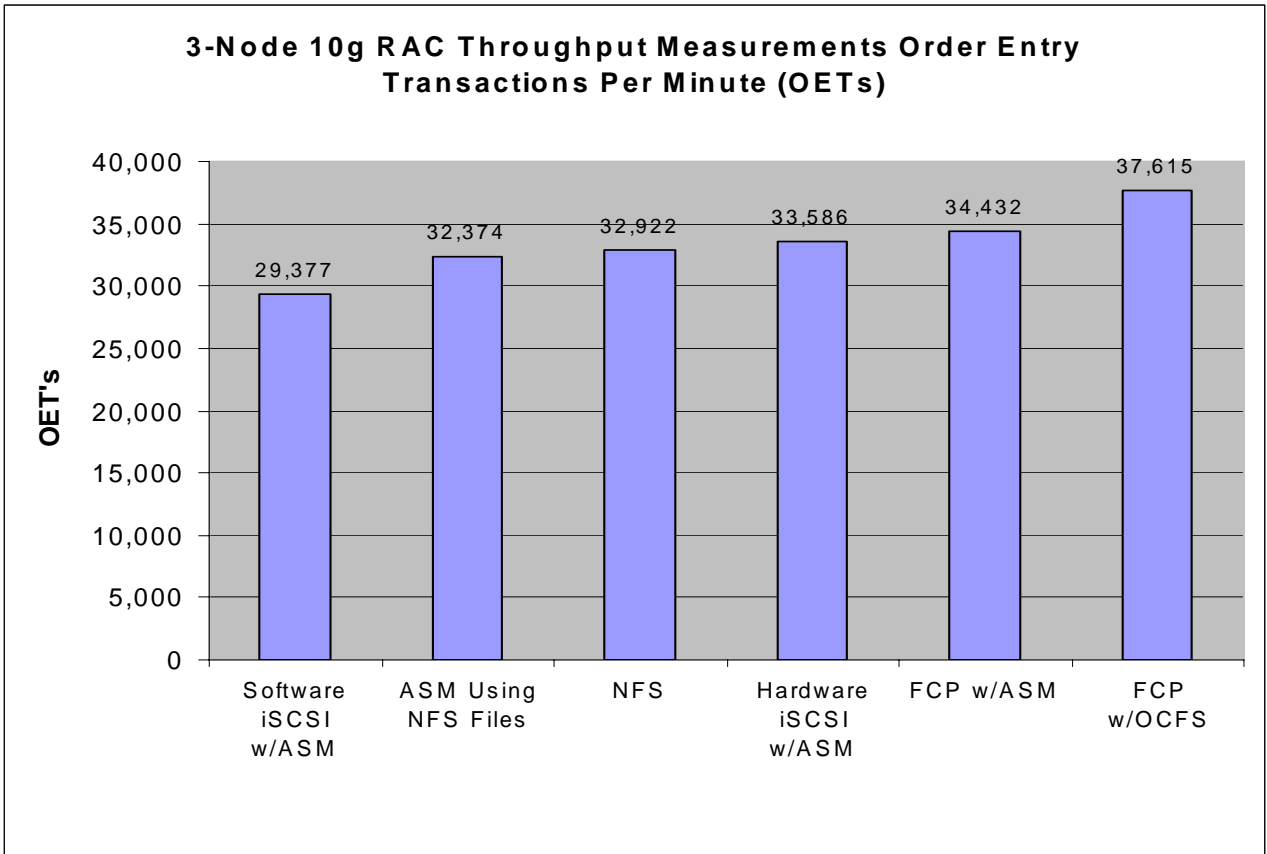


Figure 9) Overall Database Throughput For All Test Configurations

For applications or environments requiring raw disk storage, iSCSI with software initiators provides a solid low cost solution. In the test environment, the maximum throughput generated with the iSCSI software initiator was approximately 19% lower than with NFS alone. There are a couple of limitations imposed by the iSCSI software initiator. Presently, the software initiator for all Linux 2.4 kernels limits the per LUN queue depth to 12 and limits the total number of outstanding I/O's per initiator to 64. (See Appendix D for more information on this limitation.) Close examination of storage system LUN stats data from our tests did not indicate either limit being hit. Of course, performance may vary in different environments, and as iSCSI initiator software becomes more efficient, this should improve. The advantage of software iSCSI is cost, making it very attractive for cost-constrained environments where raw disk storage is required.

For the cost of iSCSI host bus adapters (HBAs), we generated a 14% improvement in performance compared to software iSCSI. The QLogic QLA4010 HBAs used in testing did not

have jumbo frames capability. In most database environments jumbo frames are desirable. Higher throughput may be possible in the future if this feature becomes available.

In our tests, we found that the throughput with NFS was just slightly lower than that of ASM with QLogic iSCSI HBAs, making it a very attractive protocol for RAC data storage. NFS on NetApp storage systems provides an excellent choice for the shared storage required by RAC databases, with very low total cost of ownership and very simple administration.

In addition to NFS as a stand alone file system, NFS files can also be used instead of raw LUNs with Oracle ASM, and provide performance just slightly less than stand alone NFS. For obvious cost reasons we have seen a good deal of interest in that environment. One potential problem with this configuration is the fact that the Oracle ASM instance defaults to the use of asynchronous I/O. At this time, asynchronous I/O (AIO) is not supported by NFS in Linux. Efforts to disable AIO by setting the Oracle parameter “disk_asynch_io” to “false” resulted in failure of the ASM instance to start up. Similarly, changing the “filesystemio_options” to “directio” will cause the instance to fail at startup as well. (You’ll recall that “directio” is a “Best Practices” requirement for Oracle10g RAC databases running on Linux.) While this does not pose a problem to regular database I/O activity, it could pose a performance problem for certain ASM operations, including the following:

- ASM rebalancing operations, in which data is redistributed across ASM disks (files). This operation is performed whenever disks (or files) are added to or removed from ASM disk groups. This was not tested during these exercises, so it is not known what the performance impact might be. A workaround for this is to defer the rebalancing operation until a period of time during which performance degradation is tolerable.
- Anytime a change is made to a disk group resulting in the writing of disk and disk group metadata. Depending on the size of the modification and performance requirements, this may or may not pose a problem.

Regardless of which protocol or configuration is chosen, Oracle10g Real Application Clusters performed extremely well with Network Appliance storage systems in the Linux environment.

During the testing, it was obvious that database throughput was strongly related to database server CPU utilization, particularly CPU utilization for user processes. Figure 10 below shows the CPU utilization observed for all configurations tested. Given the difference between configurations, it should be safe to conclude that the software iSCSI initiator environment was the least efficient and that the FCP environments were the most efficient in terms of database host “system” CPU utilization (and in terms of database performance). Simply stated, higher system CPU utilization always results in lower available CPU resources for user processes.

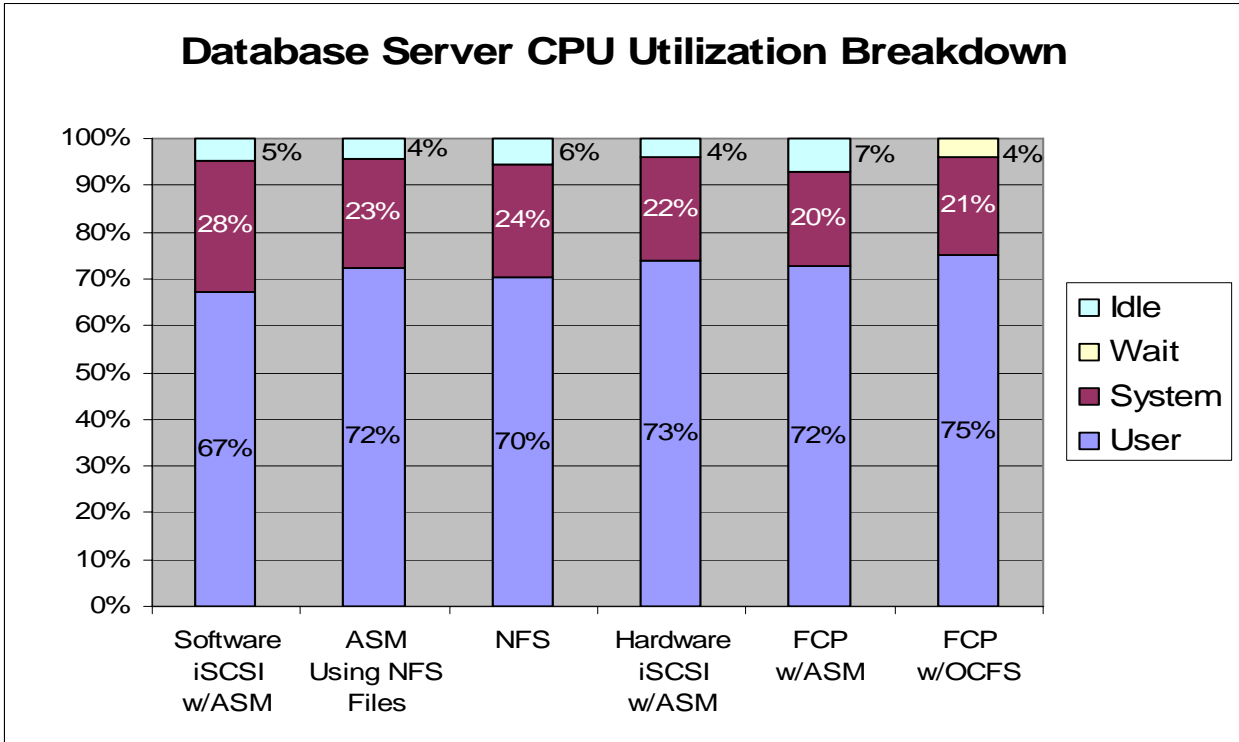


Figure 10: CPU Utilization Comparison Across Configurations

Conclusion

Network Appliance storage systems provide the ideal storage solution for Enterprise Oracle10g RAC databases. Features include:

- Scalability to grow with the enterprise
- Complete set of backup and recovery features, products and tools
- Integration with all UNIX, Linux and Microsoft® platforms
- Complete set of disaster recovery features, products and tools
- Low total cost of ownership
- Data retention products and tools
- Data security
- 24 x 7 reliability and support
- Support of NFS, iSCSI and FCP
- SAN integration

As demonstrated in these performance tests, NetApp storage systems provide the high performance required for enterprise databases and can meet the needs of any organization requiring Oracle10g RAC data storage. Available protocols include the following:

- Low cost software iSCSI for those configurations requiring block-based storage where costs must be constrained
- NFS with its low cost and ease of administration
- iSCSI with hardware HBAs
- FCP for those applications requiring high-end performance and/or SAN integration

In terms of low cost, ease of administration, solid performance and economical use of server resources, NFS is probably the best choice; however, the entire range of storage protocols listed

above is available and fully supported. Network Appliance fully supports and integrates seamlessly with the Oracle low-cost storage initiative.

APPENDIX A – LOWERING THE SGA ATTACH ADDRESS

The procedure for lowering the SGA attach address for Oracle10g R1 follows:

1. `cd $ORACLE_HOME/rdbms/lib`
2. `cp ksms.s ksms.s_orig`
3. `genksms -s 0x15000000 >ksms.s`
4. `make -f ins_rdbms.mk ksms.o`
5. `make -f ins_rdbms.mk ioracle`
6. increase shmmx to 3000000000

The above steps should be performed as follows:

- From a Linux prompt
- Logged in as the Oracle user
- On all database nodes unless a shared Oracle Home is being used
- The affected Oracle instance(s) **MUST** be shut down first

This procedure was taken from Oracle Metalink Note: 262004.1.

APPENDIX B – HOW TO ENABLE ASYNCHRONOUS I/O IN ORACLE

The information presented in this appendix was taken from Oracle Metalink Note: 225751.1.

The first step is to verify that the Linux libaio packages (libaio and libaio-devel) are installed. For the tests described in this paper, the following RPMs were installed:

- `libaio-0.3.96-5.i386.rpm`
- `libaio-devel-0.3.96-5.i386.rpm`

By default, asynchronous I/O (AIO) is disabled in Oracle. To use AIO with Oracle Databases the Oracle binary must be relinked with AIO enabled. Below is the procedure for doing this:

- `cd $ORACLE_HOME/rdbms/lib`
- `make PL_ORALIBS=-laio -f ins_rdbms.mk async_on`

These commands must be executed while logged in as the Oracle user with the affected instance(s) shut down. This should be done on all RAC nodes unless a shared Oracle Home is used.

After the above steps have been completed, set the following Oracle parameters in each instance and start the database:

- `disk_asynch_io=true`
- `filesystemio_options=asynch`

APPENDIX C – LINUX KERNEL PARAMETER SETTINGS ADDED TO /etc/sysctl.conf FILE

Parameter	Setting	Description	Additional Notes
kernel.shmax	4000000000	Max size of shared memory segment in bytes	Required for Oracle
kernel.shmall	3282294	Max amount of shared memory in pages	Required for Oracle
kernel.shmmni	4096	Max number of shared memory segments system-wide	Required for Oracle
kernel.sem	1000 32000 100 142		Required for Oracle Sets semmsl, semmns, semopm, semmni
semmsl	1000	Max number of semaphores per set or identifier	Required for Oracle
semmns	32000	Max number of semaphores system-wide	Required for Oracle
semopm	100	Max number of operations per semaphore call	Required for Oracle
semmni	142	Max number of semaphore identifiers	Required for Oracle
fs.file-max	327679	Max number of file-handles the Linux kernel will allocate	Required for Oracle
net.ipv4.ip_local_port_range	1024 65000	Local port range used by TCP and UDP	Required for Oracle
kernel.msgmnb	65535	Max number of bytes per message queue	Recommended for Oracle
kernel.msgmni	2878	Max number of message queue identifiers system-wide	Recommended for Oracle
kernel.msgmax	8192	Max size of an entire message	Recommended for Oracle
net.core.rmem_max	262144	Max receive window size	Improve network performance for NFS and RAC node interconnect
net.core.wmem_max	262144	Max transmit window size	Improve network performance for NFS and RAC node interconnect
net.core.rmem_default	262144	Default receive window size	Improve network performance for NFS and RAC node interconnect
net.core.wmem_default	262144	Default transmit window size	Improve network performance for NFS and RAC node interconnect
net.ipv4.tcp_rmem	4096 87380 8388608	Memory reserved for TCP receive buffers	Improve network performance
net.ipv4.tcp_wmem	4096 65536 8388608	Memory reserved for TCP send buffers	Improve network performance
net.ipv4.tcp_mem	4096 4096 4096	Max total TCP buffer-space allocatable	Improve network performance

APPENDIX D – QUEUE DEPTH LIMITATIONS FOR SOFTWARE ISCSI INITIATORS

The queue depth limitation described in the “Summary” section above has been resolved in SourceForge iSCSI driver version 4.0.1.10 for the Linux 2.6 kernels as follows:

- The default queue depth has been increased from “12” to “32”
- The ability to dynamically change queue depth has been added
- The total number of I/Os the driver can queue has been increased from “64” to “1024”

A partial fix is anticipated to be included in SourceForge driver versions following 3.6.2 for the Linux 2.4 kernels. That partial fix will only include the single LUN default queue depth increase from 12 to 32. For more information on this see SourceForge bug ID 759484.

APPENDIX E – NFS MOUNT OPTIONS

The NFS mount options used in these test environments are consistent with NetApp Best Practices for Oracle10g RAC databases using NFS, detailed in TR 3339 <http://www.netapp.com/library/tr/3339.pdf>.

APPENDIX F – RELEVANT ORACLE PARAMETER SETTINGS

Oracle Parameters – Non-Default Settings						
	ASM w/Software iSCSI	ASM w/NFS Files	NFS Only	ASM w/Hdwr iSCSI	ASM w/FCP	OCFS w/FCP
sga_max_size	2634022912	2634022912	2634022912	2634022912	2634022912	2634022912
sga_target	2634022912	2634022912	2634022912	2634022912	2634022912	2634022912
db_cache_size	2365587456	2365587456	2365587456	2365587456	2365587456	2365587456
db_block_size	8192	8192	8192	8192	8192	8192
parallel_max_servers	0	0	0	0	0	0
db_writer_processes	2	8	4	2	2	2
disk_asynch_io	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE
filesystemio_options	ASYNCH	DIRECTIO	DIRECTIO	ASYNCH	ASYNCH	ASYNCH
parallel_threads_per_cpu	1	1	1	2	2	2

APPENDIX G – COMMENT ON JUMBO FRAMES

Since the writing of this paper it has been reported that an MTU setting higher than 8000 may degrade network throughput in some Linux environments. Even though this behavior was never observed in the tests documented herein, caution should be exercised when enabling Jumbo frames in the context of Oracle Databases running on Linux. Jumbo frames should be treated as a tunable whereby the overall impact may vary depending upon the database environment and the nature of the database load.

