Technical report:

OracleRAC10*g* on Linux with IBM System Storage N series

A performance comparison against FCP, NFS, and iSCSI

Document NS3423-0

October 3, 2007

Table of contents

Abstract....4

Introduction....4

Executive summary....5

Workload description....5

Results summary....7

Figure 1) Overall Database Throughput Comparison For All Test Configurations....7

Performance tuning....8

Section 2: Performance Tuning....8

Establish a baseline....8

Increase the size of Oracle SGA....8

Additional Oracle performance-tuning tasks....8

Figure 2) Cumulative Impact Of Each Incremental Change....10

Test configurations....11

Section 3: Test Configurations....11

Test 1: OracleRAC10*g* with ASM using iSCSI LUNs and software initiators....11

Test Configuration 1: Oracle10*g* RAC With ASM Using iSCSI LUNs And Software Initiators....11

Storage system environment:....11

Server environment:....11

Figure 3) Network And Hardware Configuration Of RAC Environment With ASM, iSCSI LUNs And iSCSI Software Initiators....12

Oracle parameter settings....12

Network configuration....12

Test 2: OracleRAC10*g* with ASM using NFS files....13

Test Configuration 2: Oracle10*g* RAC With ASM Using NFS Files....13

Table 2) CPU Utilization Comparison Between ASM With NFS And ASM With iSCSI Software Initiators....13

Storage system environment:....13

Server environment:....13

Oracle parameter settings....14

Network configuration....14

Figure 4) Network And Hardware Configuration Of ASM Environment Using NFA Files14

Test 3: OracleRAC10*g* with NFS as a standalone file system....15

Test Configuration 3: Oracle10*g* RAC With NFS As A Standalone File system....15

Storage system environment....15

Server environment....15

Oracle parameter settings....16

Network configuration....16

Figure 5) Network And Hardware Configuration Of NFS Environment....16

Test 4: OracleRAC10*g* with ASM using QLogic QLA4010 iSCSI HBAs....17

Test Configuration 4: Oracle10*g* RAC With ASM Using QLogic QLA4010 iSCSI HBAs....17

Storage system environment....17

Server environment: 17
Oracle parameter settings 17
Figure 6) Network And Hardware Configuration Of Oracle10*g* RAC/ASM/iSCSI with QLA4010 iSCSI HBAs 18
Test 5: OracleRAC10*g* with ASM using QLogic QLA2342 FC adapters 19
Test Configuration 5: Oracle10*g* RAC With ASM Using QLogic QLA2342 Fibre Channel Adapters 19
Storage system environment 19
Server environment 19
Oracle parameter settings 20
Network configuration 20
Figure 7) Network And Hardware Configuration Of Oracle10*g* RAC and ASM With QLA2342 Fibre Channel HBAs 20
Test 6: OracleRAC10*g* with OCFS and QLogic QLA2342 FC adapters 21
Test Configuration 6: Oracle10*g* RAC With OCFS And QLogic QLA2342 Fibre Channel HBAs 21
Storage system environment 21
Server environment 21
Oracle parameter settings 22
Network configuration 22
Figure 8) Network And Hardware Configuration Of Oracle10*g* RAC/OCFS With QLA2342 Fibre Channel HBAs 22
Results comparison and summary 23
Section 4: Results Comparison And Summary 23
Figure 9) Overall Database Throughput For All Test Configurations 23
Figure 10: CPU Utilization Comparison Across Configurations 25
Conclusion 26
Conclusion 26
Appendix A: Lowering the sga attach address 27
Appendix A – Lowering The SGA Attach Address 27
Appendix B: How to enable asyncronous I/O in Oracle 28
Appendix B – How To Enable Asyncronous I/O In Oracle 28
Appendix C: Linux kernel parameter settings added to /ETC/SYSCTL.CONF FILE 29
Appendix D: Queue-depth limitations for software iSCSI initiators 30
Appendix D – Queue Depth Limitations For Software iSCSI Initiators 30
Appendix E: NFS mount options 31
Appendix E – NFS Mount Options 31
Appendix F: Relevant Oracle parameter settings 32
Appendix F – Relevant Oracle Parameter Settings 32
Appendix G: Comment on jumbo frames 33
Appendix G – Comment On Jumbo Frames 33
Trademarks and special notices 34

Abstract

A series of tests were conducted to demonstrate the high level of performance that can be obtained from an OracleRAC10g database using an IBM System Storage N series. Test results, presented in this report clearly support IBM N series storage systems as high performing, low-cost, feature-rich storage for OracleRAC10g databases.

Introduction

Recently a series of tests were conducted to demonstrate the high level of performance that can be obtained from an Oracle10*g*™ real application cluster (RAC) database (OracleRAC10*g*) using IBM® System Storage™ N series. Six different 3-node RAC configurations were tested using network file system (NFS) protocol, internet small computer system interface (iSCSI) protocol, and fibre channel (FC) protocol, running on IBM N series 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 IBM N series storage systems as high performing, low-cost, feature-rich storage for OracleRAC10*g* databases. Before continuing with the details of this study, the advantages of the OracleRAC10*g* database platform are considered.

In terms of data management, OracleRAC10*g* provides the scalability, availability and data protection required in today's 24x7 world of enterprise computing. Key features of OracleRAC10*g* 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 central processing unit (CPU) count.

Another important feature of Oracle10*g* that 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 logical unit numbers (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 results in the redistribution of data across the disks in a disk group for optimization of storage performance.

Oracle RAC requires 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. IBM N series 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, IBM N series storage systems provide superior cost-effective data protection, backup and recovery, availability, and administration through IBM N series tools and features which include the following:

- Fast reliable backups using IBM System Storage N series with Snapshot™, IBM System Storage N series with SnapVault® and IBM System Storage N series with NearStore® technologies
- Cloning and database refresh tools
- IBM System Storage N series with SnapManager® for Oracle backup and recovery
- Disk redundancy through redundant arrays of inexpensive disks (RAID) and IBM System Storage N series with RAID-DP™ (double parity)
- Storage system redundancy through the use of cluster technology
- Extensive array of disaster recovery (DR) tools.

Executive summary

The OracleRAC10*g* 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. The Linux operating system was chosen 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. The following OracleRAC10*g* storage configurations were tested:

- ASM using iSCSI LUNs with software initiators
- ASM using iSCSI LUNs with QLogic HBAs
- ASM using FCP LUNs
- ASM using NFS files as ASM disks
- NFS mounted volumes
- OCFS using FCP LUNs.

The test environments were designed 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, resource bottlenecks were eliminated in other areas including physical memory, storage and network bandwidth by connecting the three clustered systems using a Gigabit Ethernet switch and providing multiple IBM N series storage systems on which to store the database files.

Workload description

The database used for testing can best be described as online transaction processing (OLTP) in nature with a physical size of approximately 350 GB. For the testing, a set of scripts and executables were used 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 emulated the real life activities of a wholesale supplier order processing system where 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 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 RAC 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 is there I/O between the database hosts and storage systems, but there is also I/O between instances. It should also be noted that the interconnect between RAC nodes uses the user datagram protocol (UDP) network protocol instead of transmission control protocol (TCP). For these reasons, all servers in these 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, a sufficient number of consecutive test runs were executed 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 OracleRAC10*g* 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, IBM N series has several high-performance, feature-rich options for OracleRAC10*g* data storage, giving the enterprise CIO enough choices to effectively deploy IBM N series storage systems in a configuration that is consistent with the organization's goals, while benefiting from the value-add which IBM N series provides. Figure 1 shows the summary test results.

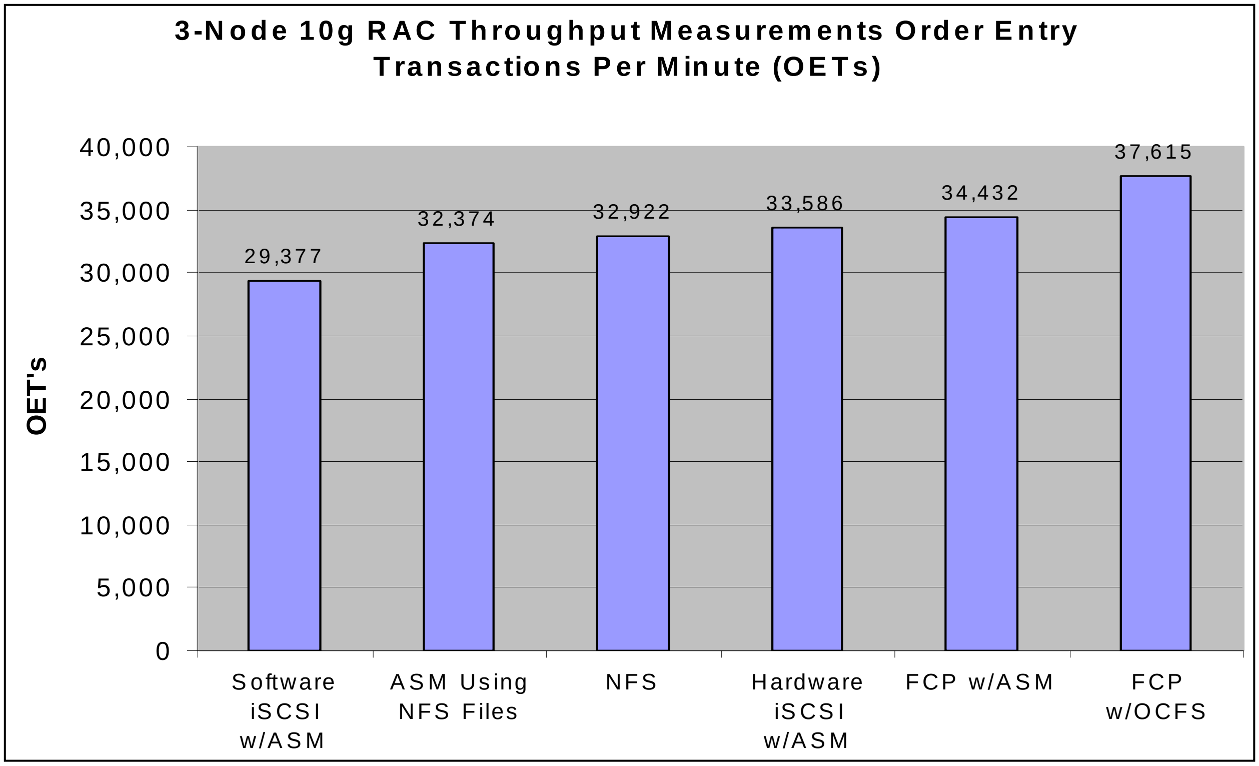

Figure 1. Overall database throughput comparison for all test configurations.

Performance tuning

Establish a baseline

To ensure uniformity across test environments, a base configuration was created to serve as a starting point for each environment to be tested. A series of preliminary tests was then conducted to improve the performance of the base configuration by modifying a number of Oracle database and Linux kernel parameters from their default values. First, a number of publicly available best practices papers were reviewed as well as general Oracle configuration guidelines. Next, a series of tuning iterations was conducted in which specific configuration parameters were modified and the resulting performance improvement measured. As a result of this tuning effort, performance improved approximately 78 percent, going from 16,484 OETs using default configuration parameters to 29,377 OETs using the tuned configuration. The rest of this section provides the details of the tuning operations that was performed.

Increase the size of Oracle SGA

The first step was to create a test database using the Oracle DBCA (Database Configuration Assistant) with default settings for all Oracle initialization parameters. The decision was made to create the OracleRAC10*g* environment using Oracle ASM 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, the size of the Oracle SGA was increased 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, host CPU utilization was observed in the 60-70% range with no indication of physical memory being a bottleneck. Additionally, no bottlenecks were observed on the IBM N series filers 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, a considerable imbalance was observed 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, 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, the scheduling priority of LMS was increased 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, another 4% improvement was achieved by increasing the db_cache_size parameter from “0” to “2256M”. The default setting of “0” enables the Oracle10*g* automatic shared memory management (ASMM) process to automatically adjust the size of the default buffer pool on the fly, based on input from the Oracle10*g* 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 and the impact of each on database throughput.

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

Table 1. Tuning details with performance impact.

Figure 2, on the following page, provides a graphical representation of this same data.

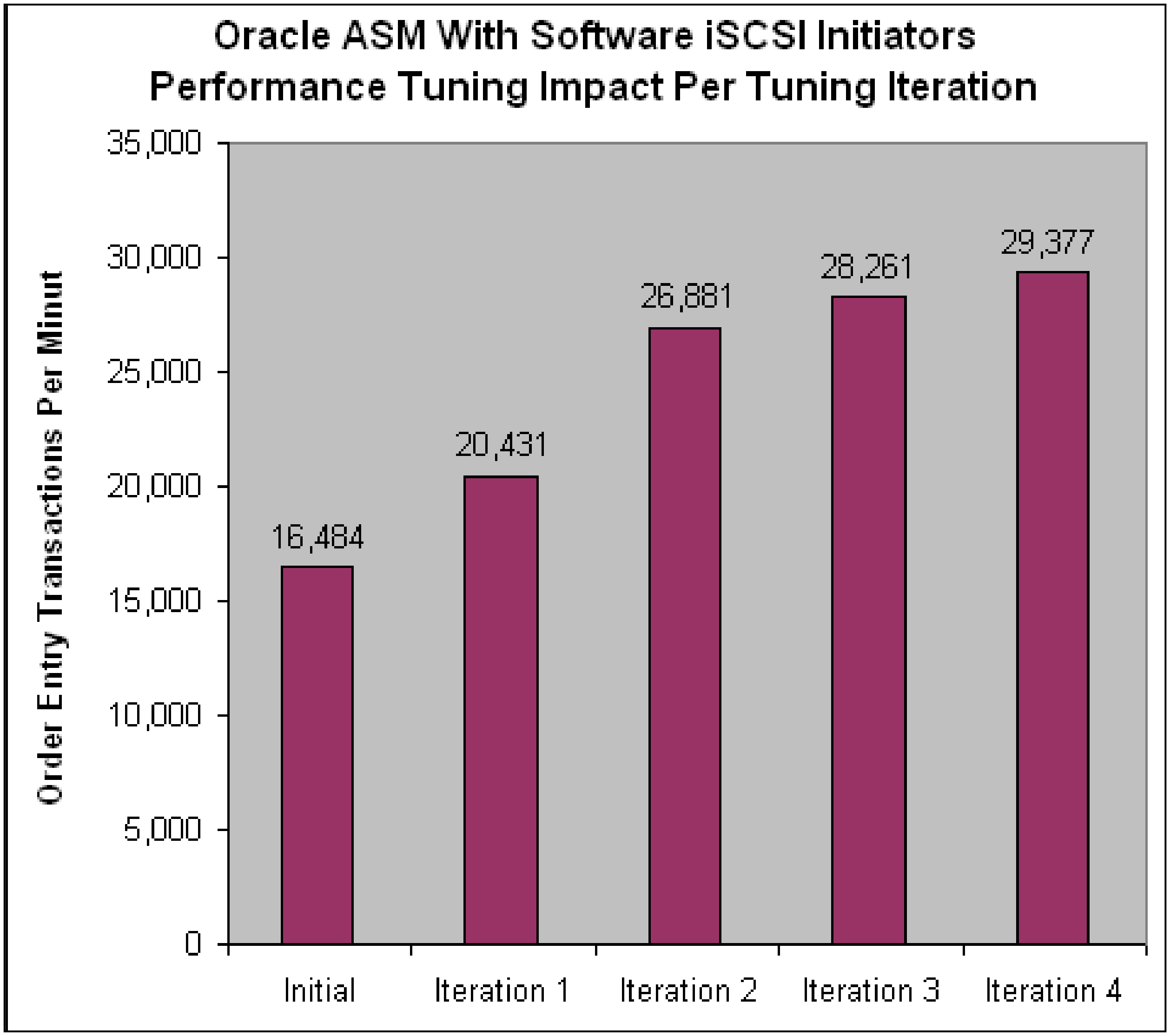

Figure 2. Cumulative impact of each incremental change.

Test configurations

This section provides the details for each of the six configurations that were 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 1: OracleRAC10*g* 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. Asynchronous I/O (AIO) was also used. 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, the Oracle executable was recompiled, 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, 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) IBM N series with Data ONTAP® 7.1 or later operating system
- (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
- (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)
- oracleasmlib-2.0.0-1
- Linux kernel parameter settings listed in Appendix C

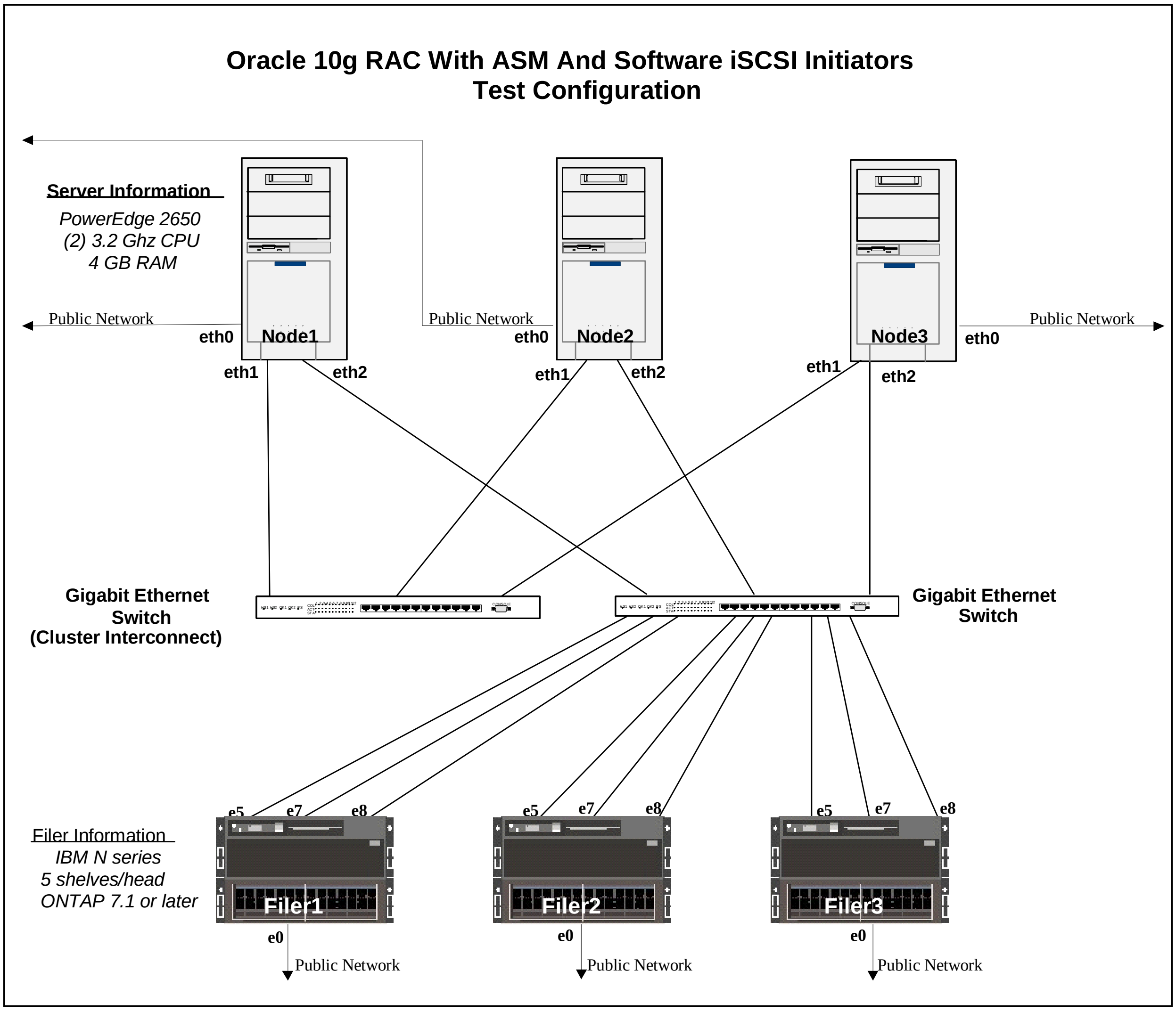

Figure 3. Network and hardware configuration of RAC environment with ASM, iSCSI LUNs and iSCSI software initiators.

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 above.

Test 2: OracleRAC10*g* with ASM using NFS files

This section details the configuration used to test a 3-node OracleRAC10*g* 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, it was determined that setting the db_writer_processes parameter to “8” yielded the highest throughput. Oracle and IBM N series best practices require the use of “directio" for RAC databases running on NFS. As a result, the parameter “filesystemio_options” was set 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 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).

Environment	CPU Usage by User Processes	CPU Usage by System Processes
Oracle ASM w/NFS Files	74%	23%
Oracle ASM w/sftwe 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) IBM N series with Data ONTAP 7.1 or later operating system
- (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
- (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,wsize=32768,tcp,actimeo=0,timeo=600

- NFS mount options used for OCR file and Voting file:
    - hard,nointr,rsize=32768,wsize=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.

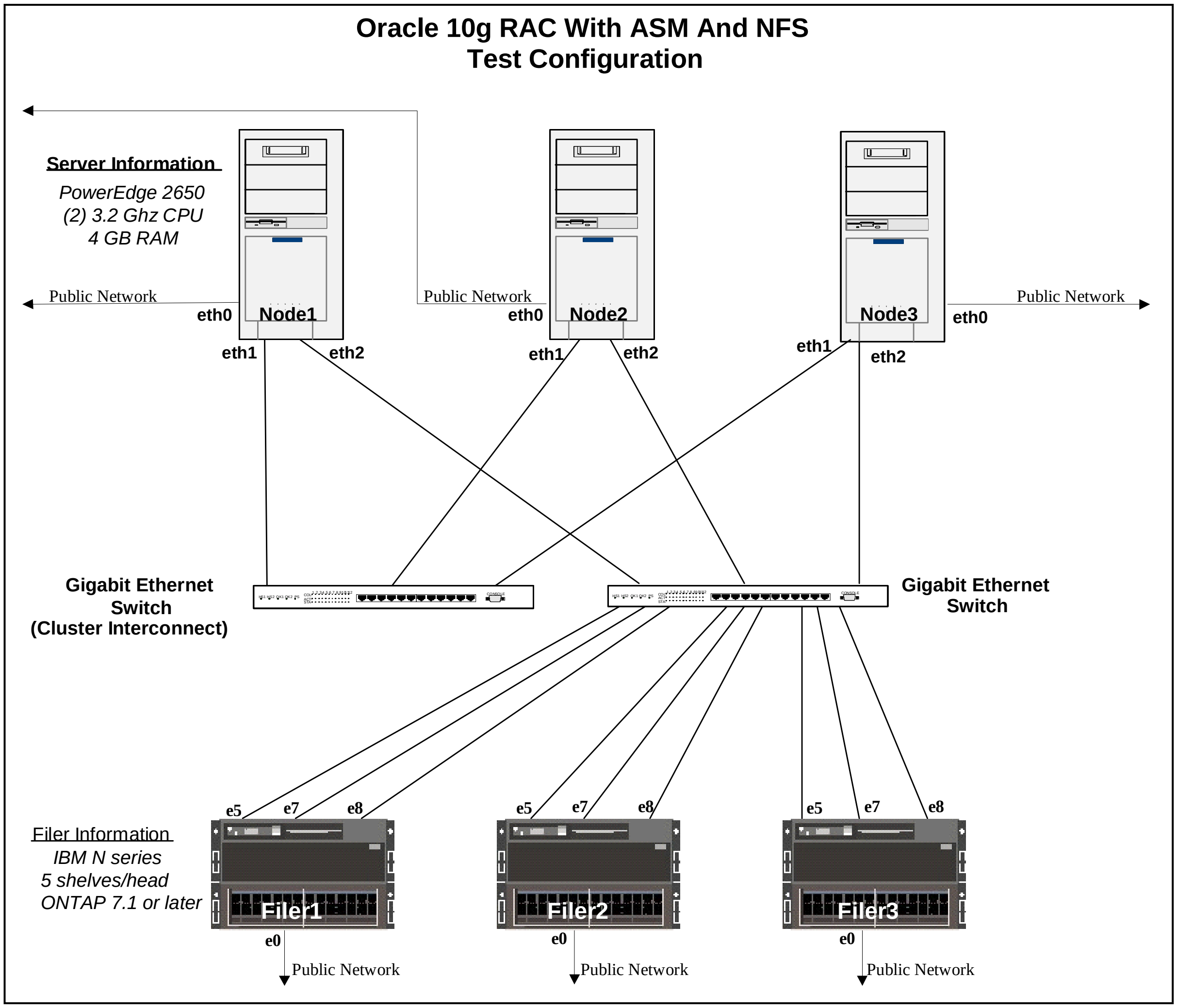

Figure 4. Network and hardware configuration of ASM environment using NFS files.

Test 3: OracleRAC10*g* with NFS as a standalone file system

This section details the configuration used to test a 3-node OracleRAC10*g* 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 it was determined that setting the “db_writer_processes” parameter to “4” yielded the highest throughput. Oracle and IBM N series 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 two 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) IBM N series with Data ONTAP 7.1 or later operating system – 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
- (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
  - Gb Ethernet 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,wsize=32768,tcp,actimeo=0,timeo=600

- NFS mount options used for OCR file and Voting file:
    - hard,nointr,rsize=32768,wsize=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.

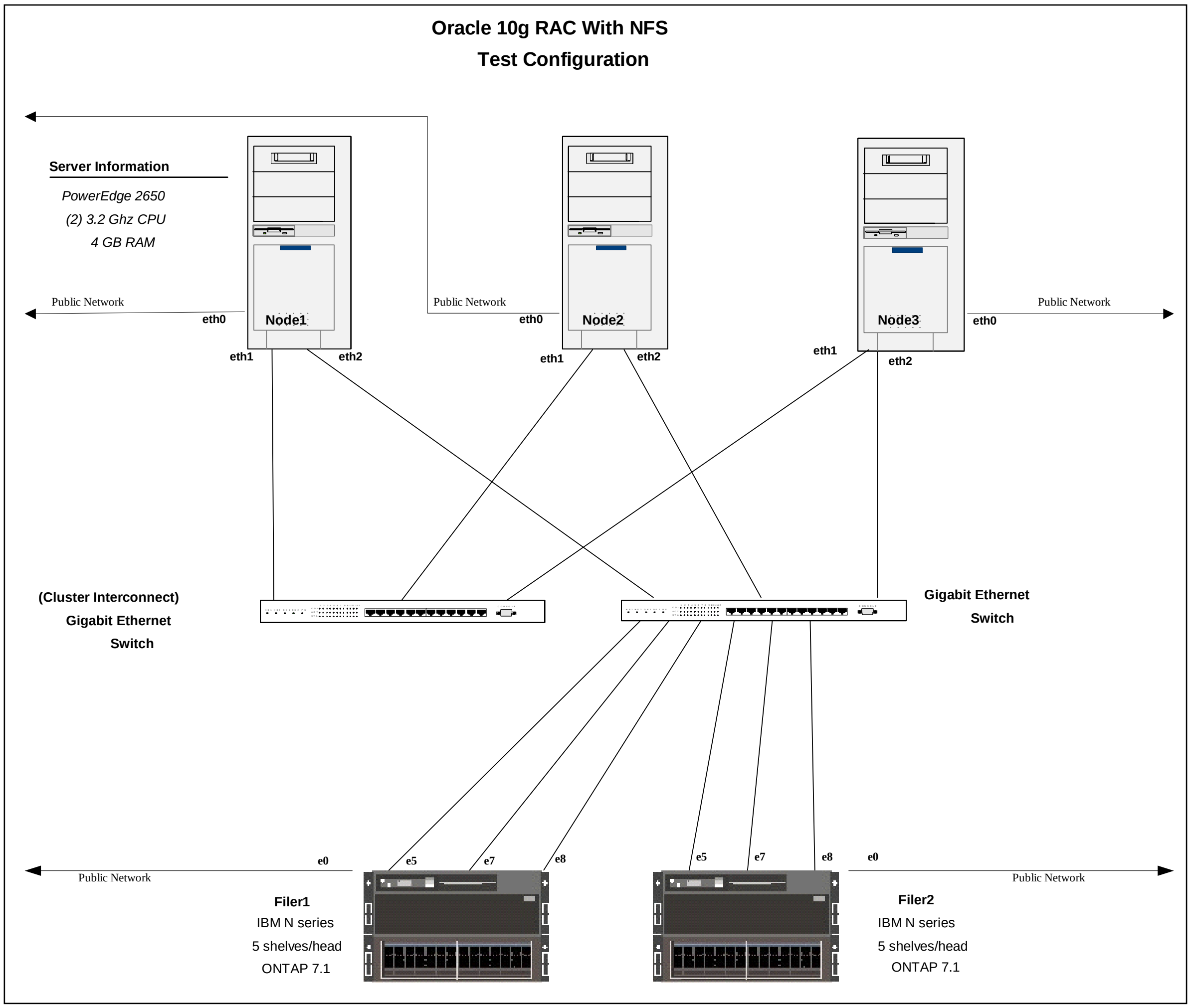

Figure 5. Network and hardware configuration of NFS environment.

Test 4: OracleRAC10*g* with ASM using QLogic QLA4010 iSCSI HBAs

This section details the configuration used to test a 3-node OracleRAC10*g* database using ASM with iSCSI LUNS and QLogic iSCSI HBAs, describes the tuning procedure and presents the results. After a few test iterations, it was 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) IBM N series with Data ONTAP 7.1 or later operating system
- (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
- (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 HBA
        - 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
    - oracleasmlib-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.

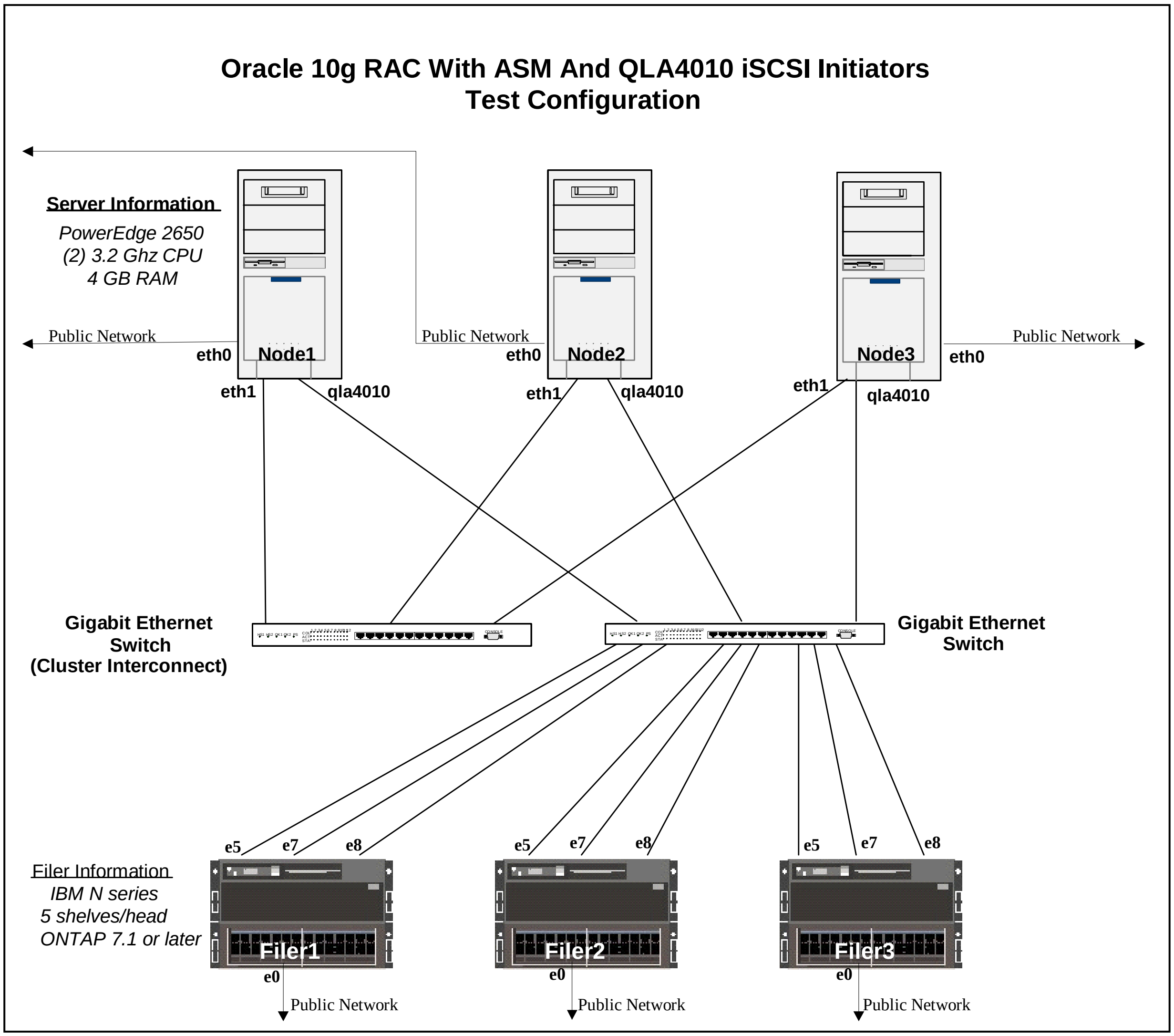

Figure 6. Network and hardware configuration of OracleRAC10*g* ASM/iSCSI with QLA4010 iSCSI HBAs**.**

Test 5: OracleRAC10*g* with ASM using QLogic QLA2342 FC adapters

This section details the configuration used to test a 3-node OracleRAC10*g* database using ASM with FCP LUNS and QLogic FCP Adapters, describes the tuning procedure and presents the results. After a few test iterations, it was 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) IBM N series with Data ONTAP 7.1 or later operating system
- (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
- (1) QLA2342 FC 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 FC HBA
    - **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
  - oracleasmlib-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.

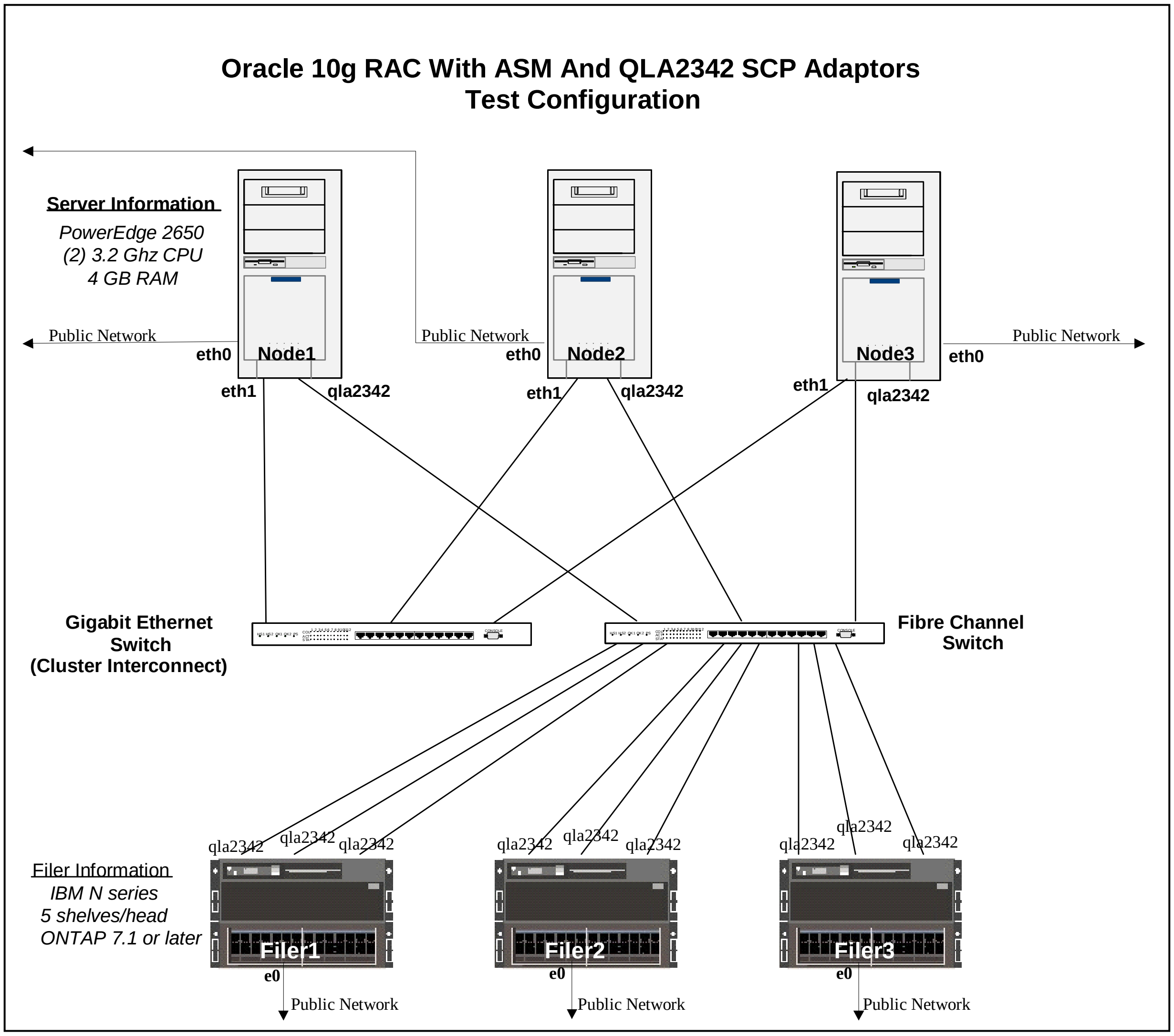

Figure 7. Network and hardware configuration of OracleRAC10*g* ASM with QLA2342 FC HBAs.

Test 6: OracleRAC10*g* with OCFS and QLogic QLA2342 FC adapters

This section details the configuration used to test a 3-node OracleRAC10*g* database using OCFS with FCP LUNS and QLogic FCP Adapters, describes the tuning procedure and presents the results. After a few test iterations, it was 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) IBM N series with Data ONTAP 7.1 or later operating system
- (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
- (1) QLogic QLA2342 FC 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 FC HBA
    - 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.

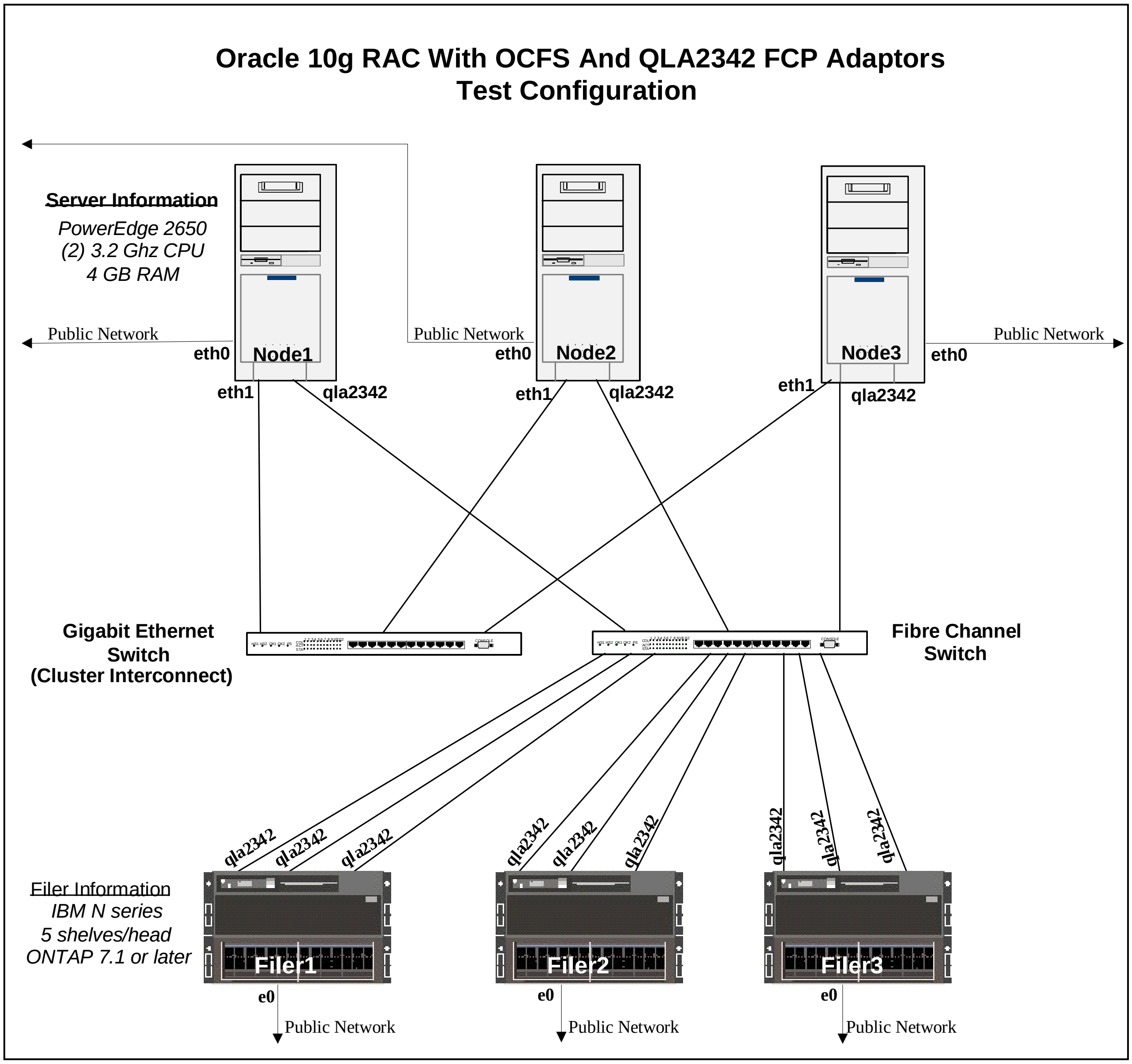

Figure 8. Network and hardware configuration of OracleRAC10*g* OCFS with QLA2342 FC HBAs.

Results comparison and summary

This final section summarizes and compares key performance test results from each of the six different OracleRAC10*g* test configurations described below. It is worth noting that, during testing, each test configuration was set up in order to stress and fully utilize the database server CPU resources in a 3-node OracleRAC10*g*. 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:

- OracleRAC10*g* with ASM using iSCSI LUNs and software initiators
- OracleRAC10*g* with ASM using NFS mounted files
- OracleRAC10*g* with NFS as a standalone file system
- OracleRAC10*g* with ASM using iSCSI LUNs and QLogic HBAs
- OracleRAC10*g* with ASM using FCP LUNs
- OracleRAC10*g* 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 FC solution is a bit costly, including the cost of FCP adapter cards and FC switches. For environments requiring FCP performance, however, it is available and performs very well.

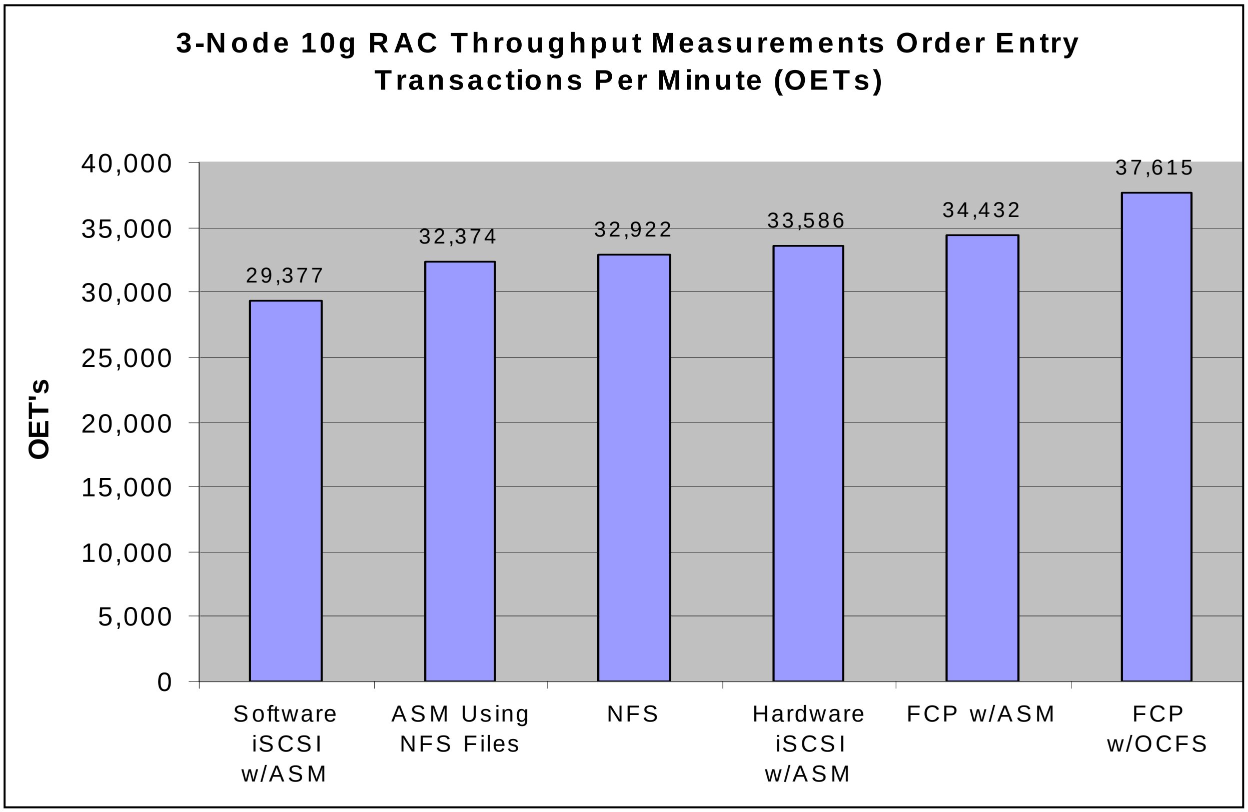

Figure 9. Overall database throughput for all test configurations.

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 these 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 HBAs, a 14% improvement in performance was achieved 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.

These tests showed 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 IBM N series 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 be used instead of raw LUNs with Oracle ASM, and provide performance just slightly less than stand-alone NFS. For obvious cost reasons, a good deal of interest has been seen 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, AIO is not supported by NFS in Linux. Efforts to disable AIO by setting the Oracle parameter "disk_asynch_io" to "false" resulted in a startup failure of the ASM instance. Similarly, changing the "filesystemio_options" to "directio" causes the instance to fail at startup,too. (Recall that "directio" is a "Best Practices" requirement for OracleRAC10*g* databases running on Linux.) While this does not pose a problem to normal database I/O activity, it could pose a performance issue 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.
- Changes to a disk group that result in the writing of disk and disk group metadata. This may or may not pose a problem (based on the modification's size and performance needs).

Regardless of which protocol or configuration is chosen, OracleRAC10*g* performed extremely well with IBM N series 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.

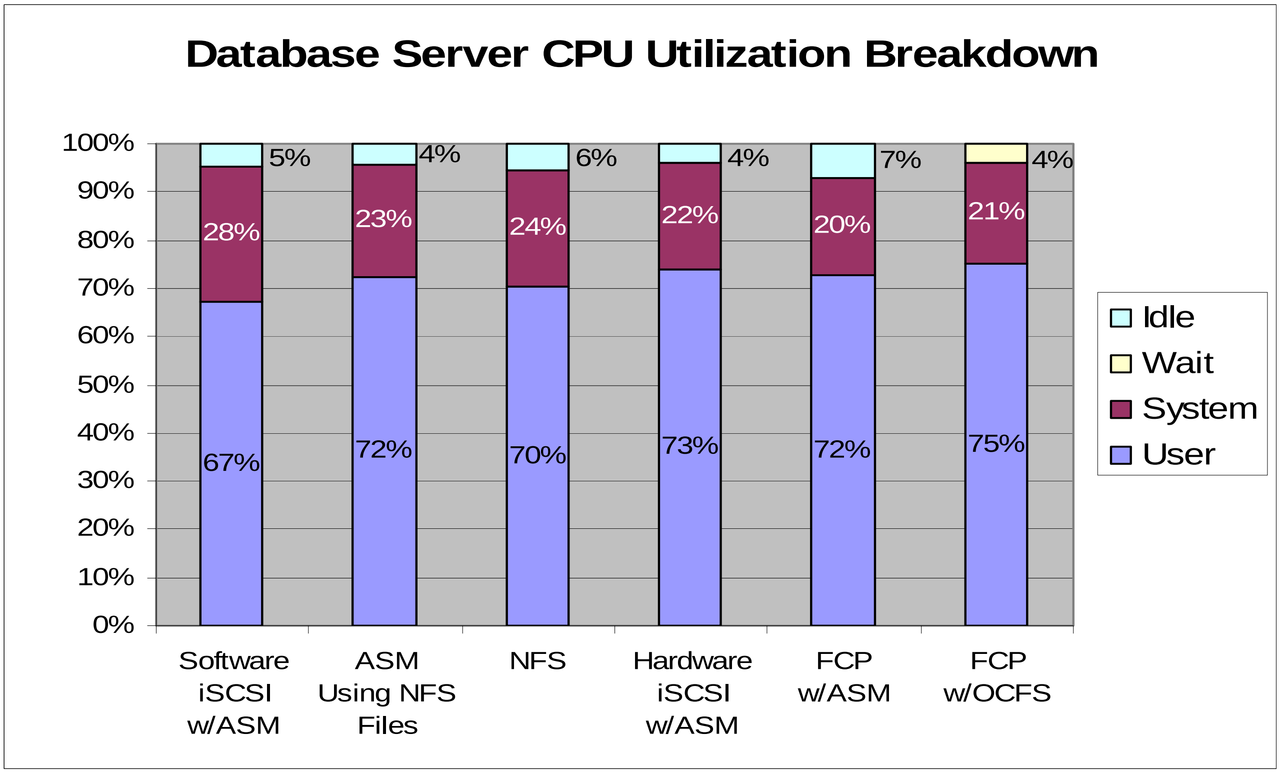

Figure 10. CPU utilization comparison across configurations.

Conclusion

IBM N series storage systems provide the ideal storage solution for enterprise OracleRAC10*g* 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 DR 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, IBM N series storage systems provide the high performance required for enterprise databases and can meet the needs of any organization requiring OracleRAC10*g* 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. IBM N series 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 Oracle10*g* 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 shmmax 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 asyncronous 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 Red Hat Packet Managers (RPMs) were installed:

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

By default, 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 IBM N series best practices for OracleRAC10*g* databases using NFS.

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.

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Network Appliance, the Network Appliance logo, Snapshot, SnapVault, NearStore, SnapManager and RAID-DP are trademarks or registered trademarks of Network Appliance, Inc., in the U.S. and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Performance measurements as depicted herein were registered on a non-IBM production unit. IBM N series storage systems will show equal if not better performance.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.