Technical Report

A Modular Scalable Architecture for Oracle Data Warehouse Environments

John Elliott, Brandon Hoang, NetApp
March 2011 | TR-3910

ABSTRACT

This document describes a modular scalable architecture that leverages Oracle® RAC, 10GbE technology, and Oracle Direct NFS on NetApp® unified storage to support large-scale sequential workloads. The architecture combines the key leading technology components to achieve multigigabyte throughput while enabling a flexible and modular implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 EXECUTIVE SUMMARY

In the past, large-scale Oracle sequential workloads such as those found in data warehouse environments were predominantly implemented on storage area network (SAN) storage with Fibre Channel protocol.

As the industry shifts away from silo deployments and toward shared infrastructure, and technology continues to rapidly advance, the options of architecting scalable high-throughput Oracle databases extend beyond the predominant Fibre Channel technology. This change is being driven, in a large part, by the Ethernet evolution. High-bandwidth Ethernet technology such as 10GbE (with 40GbE and 100GbE in development) combined with data center bridging enhancement will enable Ethernet to capture a larger share of the market in both data center networking and input/output (I/O) transport. As a result, Ethernet-based storage continues to gain momentum in large-scale, high-performance data centers and is increasingly applied in enterprise applications.

The architecture addressed in this paper presents an Ethernet-based I/O solution that is both scalable and modular. It brings together high-bandwidth 10GbE, the scalable Oracle Real Application Clusters (RAC) database platform, the Oracle Direct NFS client, and NetApp unified storage to deliver a high-throughput database infrastructure.

2 INTRODUCTION

Large-scale data warehouse environments are no longer limited to Fibre Channel storage. The industry-standard Ethernet technology commonly used for computer networking is proving advantageous in bandwidth, cost, resiliency, and versatility as it expands its presence to I/O and storage solutions.

The overall benefit of this rapidly developing Ethernet technology is undeniable. The ability to exploit the bandwidth and flexibility of Ethernet while combining it with other best-in-class products enables us to deliver a highly competitive solution.

The solution in this document demonstrates a modular and scalable architecture that supports Oracle workloads requiring high throughput. The solution takes advantage of Oracle's unique Direct NFS (DNFS) client, the latest 10GbE technology, Oracle RAC, and the NetApp unified storage platform to enable large-scale data warehouse deployments. It inherits its values of simplicity, flexibility, and low cost by combining the Ethernet component with NetApp unified storage while providing scalable performance and throughput through the use of a modular structure.

The following topics are covered in this paper:

- Industry trend on advanced Ethernet development and Ethernet-based storage
- The Oracle Direct NFS client and its benefits
- The modular and scalable architecture of our solution
- Results and data points on performance and scalability, contrasting DNFS and standard NFS (kNFS)
- Suggested tuning and optimization of the overall solution, including 10GbE, the Oracle database, and NetApp storage

2.1 AUDIENCE

This document is intended for NetApp customers, partners, and employees who might need to architect a scalable Oracle database solution for a data warehouse environment. The target audience includes database administrators, data center managers, sales engineers (SEs), consulting sales engineers (CSEs), professional services engineers (PSEs), professional services consultants (PSCs), contracted delivery partners (CDPs), and channel partner engineers.

The reader is assumed to be fairly knowledgeable of Oracle databases, Oracle RAC, and the NFS protocol.

2.2 SCOPE

This document focuses on the implementation of a high-throughput, scalable Oracle database solution that supports large-scale sequential workloads, typically found in data warehouse environments. The solution is based on Ethernet technology and NetApp unified storage products to deliver I/O by using Ethernet-based protocols. Therefore, the proposed solution and the scope of discussion are limited to the NFS protocol, including both the standard NFS and the Oracle DNFS clients.

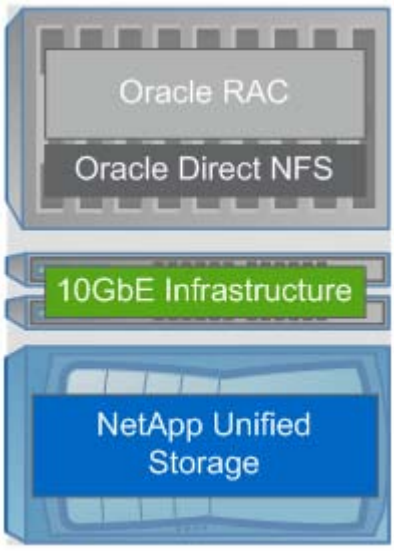
3 THE SOLUTION

As previously stated, our proposed solution is intended to be both modular and scalable, while allowing for incremental expansion as needed to support the ever-increasing workloads of successful enterprises. In addition, our configuration is composed of best-in-class components and technology to minimize complexity and cost.

The solution is made up of four enabling technologies:

- Oracle RAC
- Oracle Direct NFS client
- 10GbE technology
- NetApp unified storage

Figure 1) Solution consists of four technologies.



3.1 ORACLE REAL APPLICATION CLUSTERS

Oracle RAC is a clustered database with a shared-cache architecture that provides a highly scalable and available database solution for business applications.

Oracle RAC provides fault tolerance, performance, and scalability without any necessary application changes. It supports the transparent deployment of a single database across pools of servers, thereby providing fault tolerance from hardware failures or planned outages. It also supports mainstream business applications, including OLTP and DSS environments.

Oracle RAC provides options for scaling applications beyond the capabilities of a single server. This allows customers to take advantage of lower cost commodity hardware to reduce their total cost of ownership and provide a scalable computing environment that supports their application workload.

RAC operates on pools of servers that provide improved fault resilience and modular incremental system growth over single symmetric multiprocessor (SMP) systems. In the event of a system failure, clustering provides users with high availability. Access to mission-critical data is not lost. Redundant hardware components, such as additional servers, network connections, and disks, allow the cluster to provide high availability. Such redundant hardware architectures avoid a single point of failure and provide exceptional fault resilience.

3.2 ORACLE DIRECT NFS

The Oracle Database 11*g* Direct NFS (DNFS) client integrates traditional NFS client functionality directly into the Oracle Database software. Through this integration, Oracle can optimize the I/O path between Oracle and the NFS server, providing improved performance compared to traditional NFS implementations. In addition, the DNFS client simplifies, and in many cases automates, the performance optimization of the NFS client configuration for database workloads.

In contrast to DNFS, standard NFS client software provided by the operating system is not optimized for Oracle Database file I/O access patterns and usually incurs additional server CPU overhead at the operating system kernel.

The DNFS client is capable of performing concurrent direct I/O that bypasses all operating system-level caches and eliminates bottlenecks associated with operating system write-ordering blocks. This decreases memory consumption by eliminating scenarios in which Oracle data is cached both in the SGA and in the operating system cache. It also eliminates the kernel mode CPU cost of copying data from the operating system cache into the SGA. The Direct NFS client also performs asynchronous I/O, which allows processing to continue while the I/O request is submitted and processed.

Oracle DNFS client implements multipath I/O internally. It optimizes performance by automatically load-balancing requests across all specified network paths. If one network path fails, the DNFS client reissues commands over the remaining paths, providing fault tolerance and high availability.

With DNFS, using NFS for databases becomes even simpler. It eliminates the common problem of inconsistency in managing configurations across different platforms by providing a standard NFS client implementation across all platforms supported by the Oracle Database.

Overall, DNFS generally outperforms traditional NFS clients, is simple to configure, and provides a standard NFS client implementation across all hardware and operating system platforms.

3.3 10-GIGABIT ETHERNET TECHNOLOGY

Ethernet technology has existed for over three decades. Its continued rapid evolution resulted in the introduction of 10GbE to the market back in 2006.

Today, 10GbE provides a cost-effective way to increase network throughput and bandwidth. More specifically, it provides the necessary performance, proven flexibility, and lower-cost advantage that today's data centers require. Indeed, with continually falling prices per port, 10GbE technology and related components are fast becoming the technology of choice for the core layers of the data center as well as the I/O transport layer for enterprise storage.

In addition to higher bandwidth, the introduction of the data center bridging protocol and enhancements to IEEE standard to improve the robustness and reliability of Ethernet should further drive the demand and adoption for 10GbE. 10GbE in itself continues to facilitate the move toward converged networking and multiprotocol storage networking.

3.4 NETAPP UNIFIED STORAGE

NetApp is recognized as the industry-leading vendor in unified storage, marked by its first introduction of unified storage systems to the market nearly a decade ago. The NetApp Unified Storage Architecture provides true multiprotocol support, a single management interface, integrated data protection, support for multiple tiers of storage (primary, secondary, and archive/compliance), quality of service, and the ability to act as a front end for legacy storage systems. NetApp is able to combine these features and more into a single platform capable of meeting customers' end-to-end storage needs, while demonstrating significant performance and cost-of-ownership advantages.

The NetApp Unified Storage Architecture provides huge benefits in scalability, ease of management, increased efficiencies, and reduced storage costs.

The trend toward consolidation and server virtualization is transforming the way data centers are being designed, built, and managed. There is a need to move away from the traditional "silo" approach to a shared and flexible infrastructure. Data storage and data management are key elements of this transformation. This shift toward a shared infrastructure increases the importance of unified storage and enables customers to achieve maximum storage efficiency while improving flexibility, increasing scalability, controlling power and cooling, and much more. Unified storage is a critical component of customers' shift to a shared infrastructure.

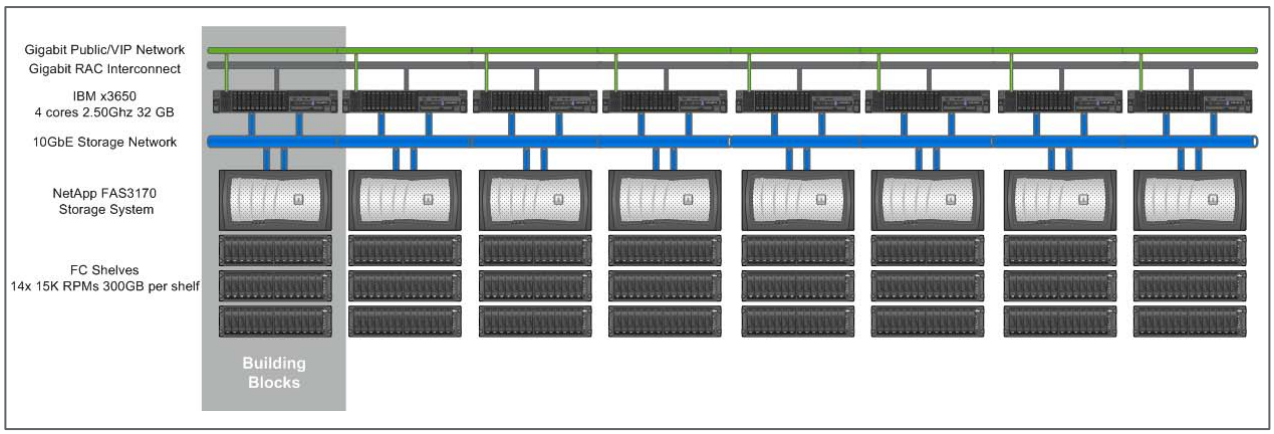
4 ARCHITECTURE

Figure 2 provides a basic graphical representation of our configuration in terms of hardware and resource utilization. The overall architecture is constructed using basic building blocks. Each block is made up of a single Linux® server running Oracle Database 11*g* R2 software and is equipped as follows:

- A single 1GbE NIC used for Oracle RAC cluster interconnect and cache fusion
- A single 1GbE NIC used for public access and RAC VIP functionality
- Two 10GbE NICs for storage access
- Storage resources equivalent to a single NetApp FAS3170 controller equipped with three shelves of FC disks, two 10GbE ports, and a single 1GbE port for public access

The use of basic building blocks adds to the effect and the definition of the modular and scalable architecture. One or more of these blocks of computing resources can be deployed to support environments of different sizes and capacity requirements. Blocks, or modules, of a standardized configuration can be added to increase scalability or to meet the workload expansion.

Figure 2) Graphical representation of test configuration in terms of hardware and resources.



4.1 HARDWARE DETAILS

Table 1 provides the specifics of the hardware used for the testing including both RAC servers, NetApp storage controllers, and the networking gear used to connect it all together.

Table 1) Hardware details.

Category	Component	Quantity	Specification
Server	Application server	1	IBM dual-core Intel® Xeon® 8GB
	RAC database servers	8	IBM X3650 quad-core Intel Xeon processor E5420 (2.5GHz 12MB L2), 32GB, Broadcom NetXtreme II BCM5708 Gigabit interfaces
Storage	Storage controllers	8	FAS3170 4 processors 32GB with dual 10G Ethernet controller T320E-XFP
	Disk shelves	24	DS14MK4 with 300GB 15K RPMs FC drives
Network	10GbE switches	2	Cisco Nexus® 5020
	1GbE switches	2	Cisco® 4948
	10GbE network adapter cards	8	Intel 10GbE server adapter X520-SR2

4.2 SOFTWARE DETAILS

Table 2 shows the software details used for the testing.

Table 2) Software details.

Category	Version
Server operating system	Red Hat Enterprise Server 5.3 64-bit
Storage operating system	NetApp Data ONTAP® 8.0 7-Mode
Oracle	Oracle Grid Infrastructure 11*g*R2 (11.2.0.1) Oracle RAC/RDBMS Enterprise Edition 11*g*R2 (11.2.0.1)
Workload	DSS type workload

4.3 NETWORK DETAILS

A RAC environment running Ethernet-based storage usually consists of three types of networks:

- **Public network.** Sometimes referred to as the application network. Clients and applications communicate with Oracle instances by using this network. Additionally, Oracle RAC VIPs are attached to the public-network interface on each database server.

- **Interconnect network.** Required for communication between RAC nodes and the transfer of cache fusion data.
- **Storage I/O network.** Enables the storage I/O transport layer between the database servers and storage systems.

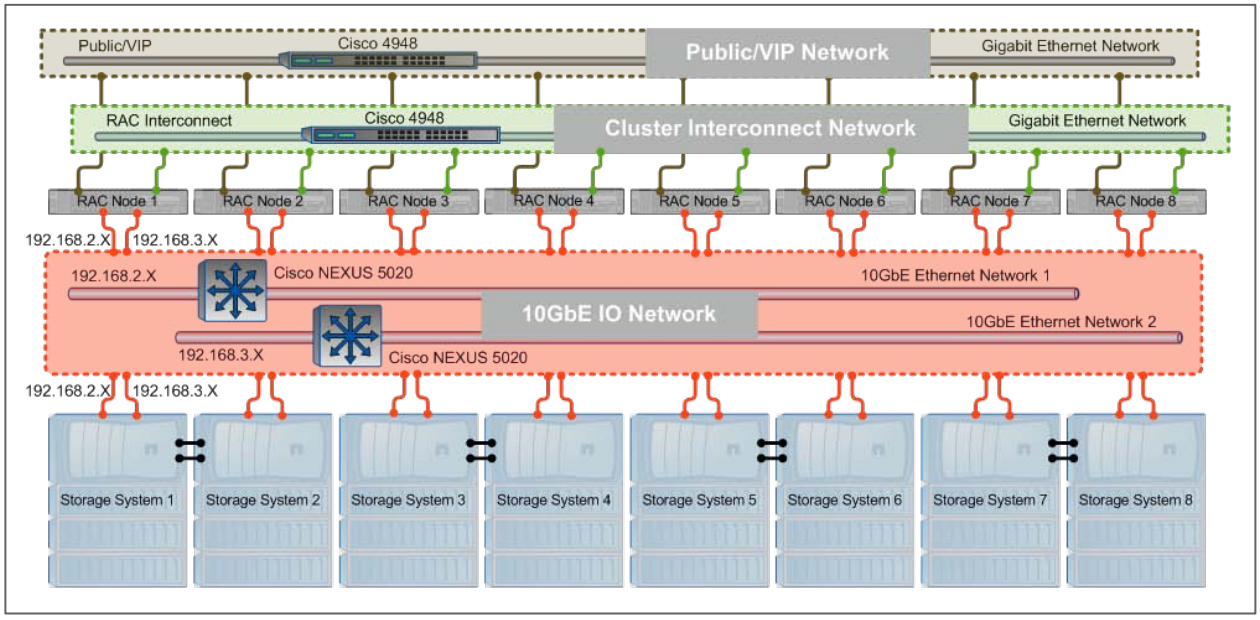
In our test infrastructure, both the public network and the cluster interconnect network are gigabit-based networks. For the storage network, we used 10GbE. Each database server and storage system was configured with two dedicated 10GbE connections for storage I/O access, and jumbo frames were enabled.

Table 3 provides details of the networks used for the testing. Figure 3 shows how each network was deployed in the overall environment. For all of the tests, we enabled jumbo frames on the 10GbE infrastructure.

Table 3) Network details.

Network Type	Network	Bandwidth	MTU
Public and VIP	10.61.172.X	Gigabit	1,500
Cluster interconnect	192.168.1.X	Gigabit	1,500
Storage and I/O	192.168.2.X 192.168.3.X	10GbE 10GbE	9,000 (jumbo frames) 9,000 (jumbo frames)

Figure 3) Network configuration.



4.4 STORAGE LAYOUT

In our test environment, efforts were made to optimize the layout of volumes and data files for performance and ease of management. Each NetApp FAS3170 storage system is configured with a single large aggregate made up of 36 disks using RAID-DP®. This configuration maximizes the use of available disk spindles for the RAID group configuration while reserving a sufficient number of disks for hot spares. Data file and log volumes of uniform size were constructed from the aggregates to provide a balanced storage layout, resulting in each storage controller hosting two data file volumes and one log volume.

To maximize network bandwidth for I/O operations and to load-balance I/Os across the two 10GbE networks, data file volumes and log volumes on the database servers were carefully distributed across the two available 10GbE connections and across the eight NetApp storage controllers. Figure 4 shows how the data file and the log volumes were laid out in terms of the storage controllers and the network.

Figure 4) Storage and volume layout.

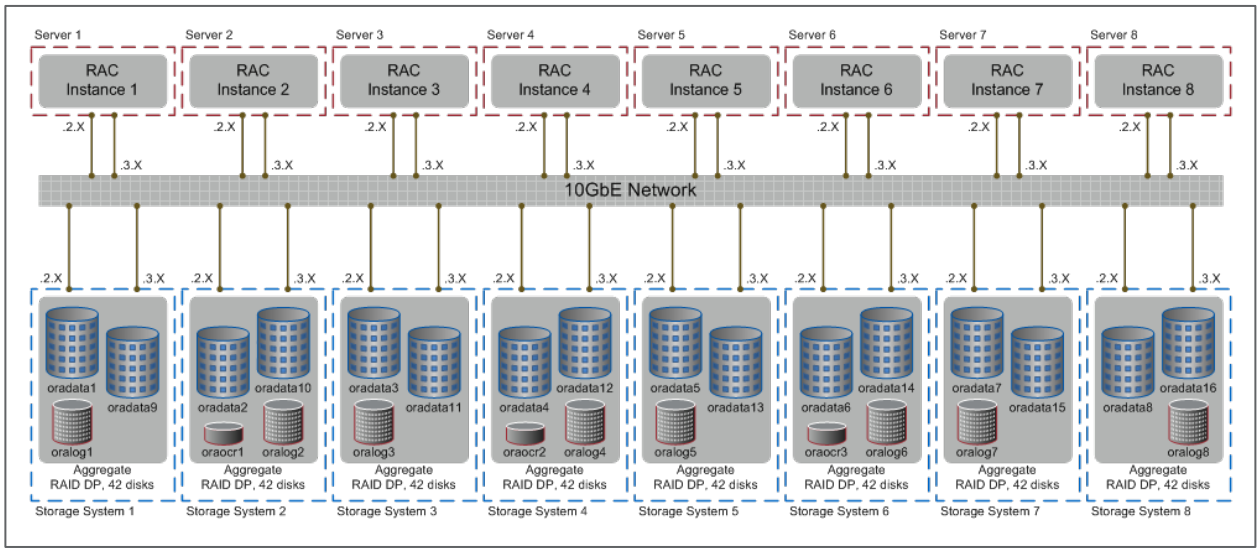


Figure 4 has a total of 16 data file volumes (oradata1 to oradata16), 8 redo log volumes (oralog1 to oralog8), and 3 ocr volumes (oraocr1 to oraocr3). Table 4 provides the total size of the different volumes created for the testing.

Table 4) Volumes and network distribution.

Mounted Network (10GbE)	Volume Type	Volumes	Size (per volume)
192.168.2.X	Data file	oradata1, oradata2, oradata3, oradata4, oradata5, oradata6, oradata7, oradata8	2.4TB
	Redo log	oralog1, oralog3, oralog5, oralog7	8GB
	OCR/voting disk	oraocr1, oraocr2, oraocr3	3.2GB
192.168.3.X	Data file	oradata9, oradata10, oradata11, oradata12, oradata13, oradata14, oradata15, oradata16	2.4TB
	Redo log	oralog2, oralog4, oralog6, oralog8	8GB

5 PERFORMANCE AND SCALABILITY RESULTS

5.1 WORKLOAD

At the most basic level, our workload can be described as a continuous stream of large sequential reads. To generate that workload, we used a series of SQL queries typically used for aggregation of data in data warehouse environments. Those queries were executed against a 3TB Oracle 11*g* RAC database designed to emulate the real-world decision support systems of a worldwide wholesale supplier. The database contained tables ranging in size from 600,000,000 to 18,000,000,000 rows, with tablespaces made up of 276 data files. We configured our database with a total of eight RAC nodes.

5.2 TEST METHODOLOGY

Our goal was to demonstrate the modular scalable architecture of Oracle RACs with DNFS and 10GbE. Therefore, we tested our database with a single active node, then with two active nodes, then with three active nodes, and so forth, all the way up to eight active nodes while applying the same workload directly to each active node. We recognized the fact that Oracle RAC databases should also scale with the Linux kernel NFS (kNFS) client and that a comparison between DNFS and kNFS could be very interesting, particularly in the areas of host CPU utilization and load balancing. With that in mind, we defined the scope of our project to include the following configurations:

- DNFS using a single 10GbE port on each Oracle server and each storage controller.
- DNFS using two 10GbE ports on each Oracle server and each storage controller, allowing DNFS to balance the workload across both 10GbE connections.
- kNFS using a single 10GbE port on each Oracle server and each storage controller.
- kNFS using two 10GbE ports on each Oracle server and each storage controller with manual load balancing across both 10GbE connections.

Note: The manual load balancing was performed at the time our database was created to take full advantage of the bandwidth of each of our eight storage systems. This was done by spreading our data files across all NFS mountpoints in round-robin fashion, resulting in a fairly even distribution of the workload across the 10GbE ports.

5.3 RESULTS

For all four of these configurations, we observed near-linear scaling for each, with the real differences in measured performance being in throughput and host-side CPU utilization. In our testing, we found that DNFS configured with 2 x 10GbE ports on all our servers and storage controllers generated a 12.8% better performance compared to kNFS in a similar configuration. Additionally, we found that DNFS and kNFS delivered comparable performance when using only a single 10GbE wire, while DNFS delivered this performance while consuming 30% less server CPU resources compared to kNFS. Figures 5-7 provide more details on how our database workload scaled for each of the four configurations.

Figure 5 compares the throughput in MB/s delivered by both kNFS and DNFS with both one and two 10GbE network interfaces configured to handle the database load.

Figure 5) Average database throughput by RAC node count.

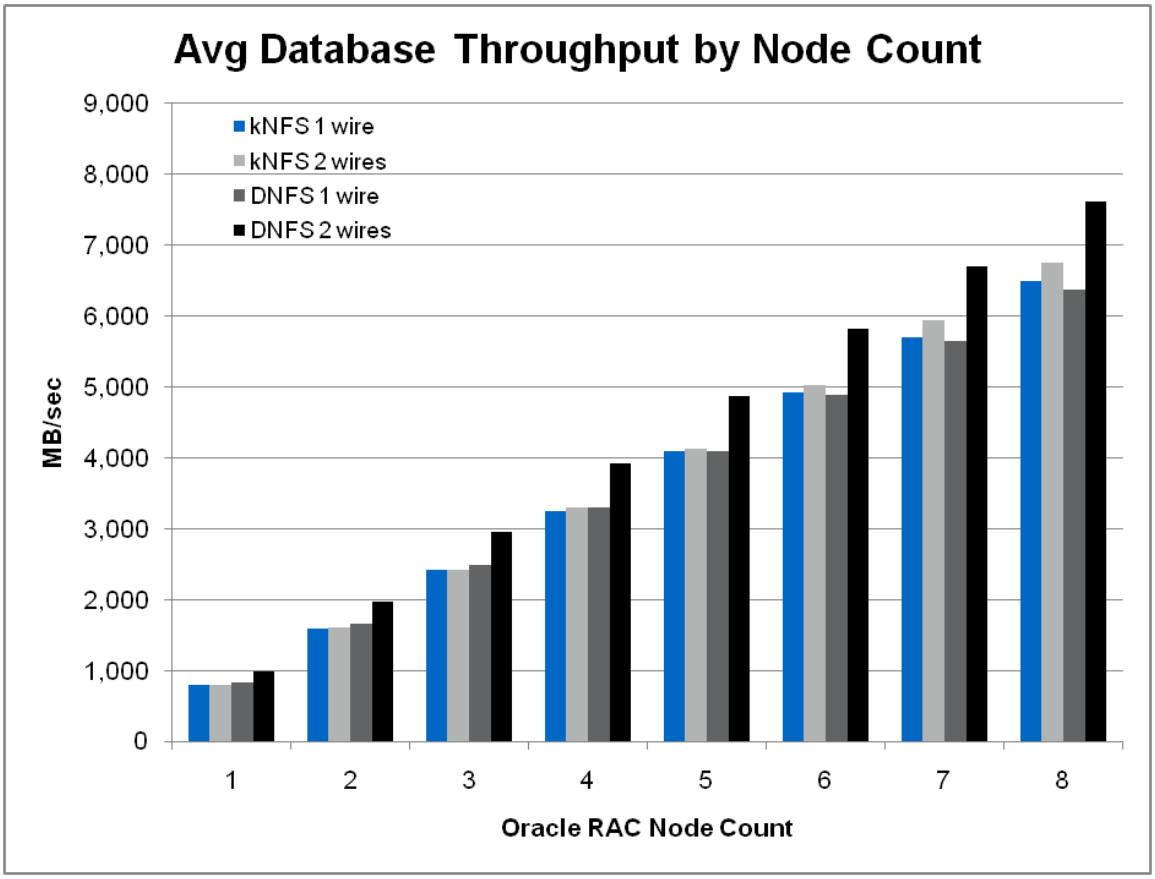


Figure 5 shows the following results:

- Workload throughput for each configuration increases in a near-linear fashion as the node count increases from 1 to 8, demonstrating the scalability and modularity of the Oracle RAC and NetApp solution. Each time we add a node, we are increasing server, network, and storage bandwidth utilization.
- Single-wire and two-wire kNFS performance remains about the same until our node count reaches 4. As we continue to ramp up to 8 nodes, the difference becomes more noticeable. With DNFS, the two-wire configuration consistently outperforms our one-wire configuration. This is the result of the DNFS client's automatic load-balancing feature.
- In terms of throughput, DNFS with two 10GbE wires outperformed the other configurations tested, as the following summary for our 8-node configuration shows:
  - 19.5% higher than DNFS with 1 wire
  - 17.3% higher than kNFS with 1 wire
  - 12.8% higher than kNFS with 2 wires
- Throughput for the two-wire kNFS configuration, in the best case, was only about 4% higher than for the one-wire configuration. Figure 7 shows that the host-side CPU was very nearly saturated for the one-wire configuration, leaving very little room for improvement with the addition of the second 10GbE port.

Figure 6 shows a different view of throughput scaling and comparison. The data represented here is the average throughput per active RAC node for each configuration tested, going from 1 to 8 nodes. As

Figure 6 shows, instance-level throughput remains very consistent as the node count increases, which once again demonstrates the scalability and modularity of our solution.

Figure 6) Average instance throughput by node count.

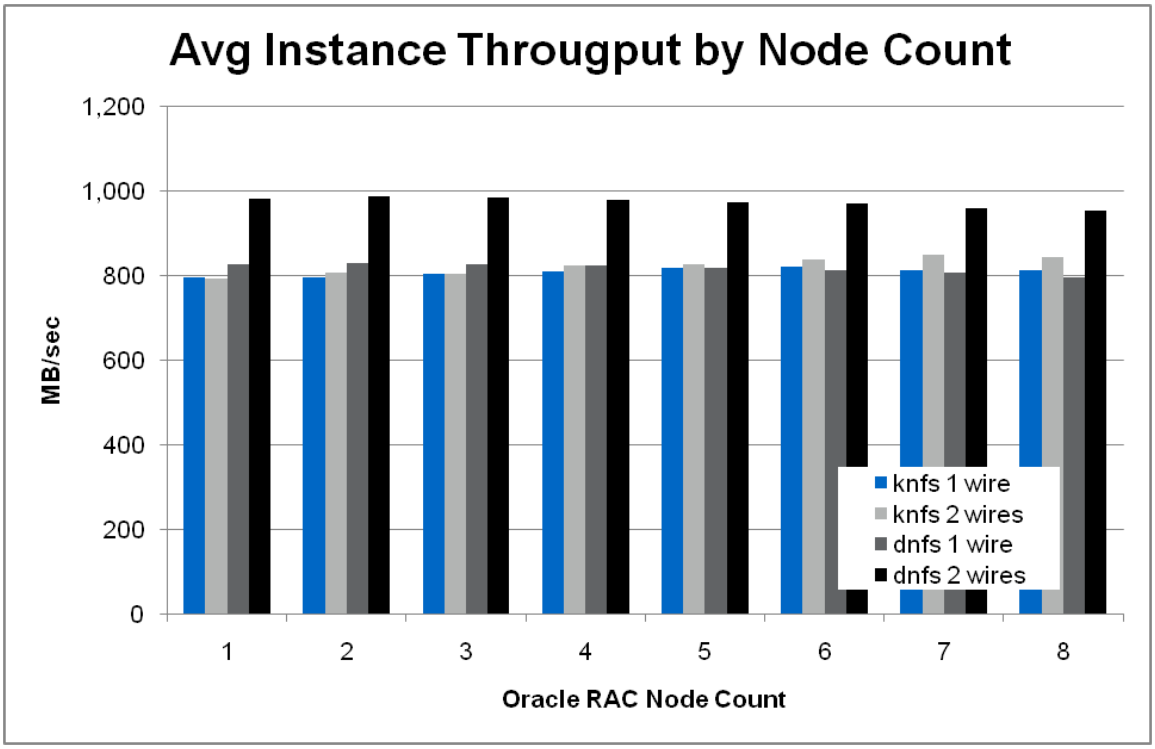
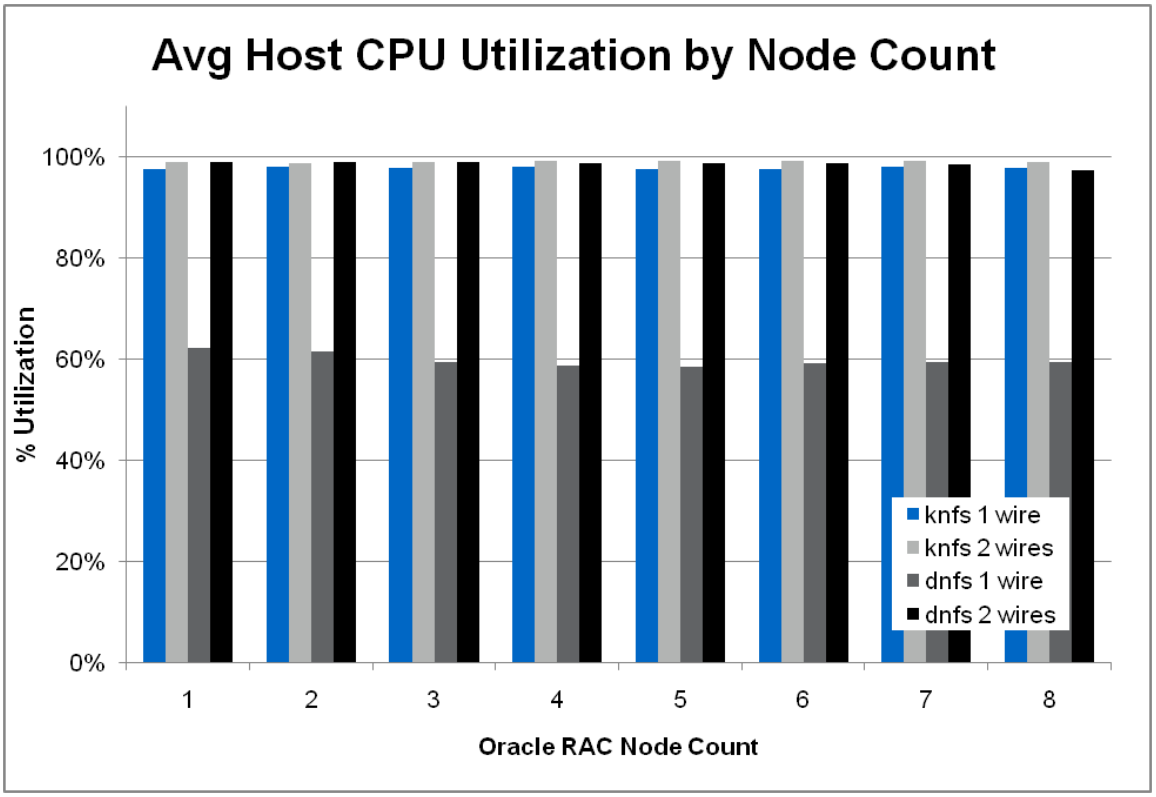


Figure 7 represents the average host CPU utilization corresponding to the throughput results presented in Figure 6. This CPU utilization data provides a better understanding of the throughput data and further demonstrates the performance improvements possible when using DNFS compared to kNFS. You'll immediately notice that DNFS using a single 10GbE wire uses approximately 30% less (~60% compared to 98%) host CPU resources compared to using kNFS and a single 10GbE connection. Clearly, using DNFS leaves more CPU bandwidth for other applications or to improve the performance of existing applications. In our case, we were able to add another 10GbE port to each node of our configuration and significantly increase our database throughput by taking advantage of the remaining host CPU resources. In this case, we were able to consume the remaining host CPU and drive higher throughput levels.

It's also easy to understand why adding a second wire to our kNFS configuration (already using 98% of host CPU bandwidth) provided only minimal increases in throughput. These observations not only further demonstrate the scalability and modularity of our solution, but also emphasize the strong contribution of DNFS to the Oracle RAC configuration. With DNFS, we are able to utilize host CPU more efficiently and achieve higher levels of throughput for our two-wire DNFS configurations, as shown in Figure 6 and Figure 7.

Figure 7) Average host CPU utilization by node count.



The data presented in the previous figures clearly illustrates the benefits of combining NetApp unified storage with Oracle RAC and DNFS for achieving scalable, modular data warehouse environments. To further validate the throughput, we examined our database performance in real time using Oracle Enterprise Manager (OEM). Figure 8 and Figure 9 contain OEM screen captures performed while our workload was being applied to our two-wire DNFS configuration. Figure 8 shows throughput for a single RAC instance quickly ramping up, approaching 1,000 MB/s, and then remaining fairly constant for the duration of our test measurement interval. This is consistent with the data presented in Figure 6.

Figure 8) Instance throughput for a single RAC node using DNFS.

Figure 9 shows the total throughput of our eight-node database approaching 8,000 MB/s and leveling off for the duration of our test measurement interval. The data represented here is consistent with the data presented in Figure 5.

Figure 9) Total database throughput for eight RAC nodes using DNFS.

6 TUNING AND OPTIMIZATION

Tuning and optimization were applied at different layers of the architecture to enhance the overall performance of the Oracle RAC database using the specified workload.

6.1 SERVER HARDWARE CONFIGURATION

For our tests, we used PCI-E x8 (8-lane) 10GbE NICs. We initially tested with the NICs installed in the PCI-E x4 (4-lane) host-side slots, and the performance fell far short of our expectations. We discovered the following messages in our Linux `/var/log/messages` file:

```
PCI-Express bandwidth available for this card is not sufficient for optimal
performance.
For optimal performance a x8 PCI-Express slot is required.
```

Moving the NICs to 8-lane slots nearly doubled our throughput. For optimal 10GbE throughput, 8-lane cards should be used, and they must be installed in 8-lane slots.

6.2 LINUX KERNEL TUNING

Shared memory and semaphores are two important resources for an Oracle instance running on any platform. Inappropriate settings of shared memory and semaphores could limit the size of the Oracle SGA and the Oracle process communication, which might lead to a significant impact on the overall database performance.

As shown in Table 5, the "Oracle Database Installation Guide 11*g* Release 2 (11.2) for Linux" recommends the following minimum settings.

Table 5) Minimum settings.

Parameter	Description
`fs.aio-max-nr = 1048576`	This value limits concurrent outstanding requests and should be set to avoid I/O subsystem failures
`fs.file-max = 6815744`	Maximum number of file handles
`kernel.shmall = 2097152`	Total amount of shared memory pages that can be used systemwide
`kernel.shmmax = 536870912`	Maximum size in bytes of a single shared memory segment

Parameter	Description
`kernel.shmmni = 4096`	Maximum number of shared memory segments systemwide
`kernel.sem = 250 32000 100 128`	• Maximum number of semaphores per semaphore set • Total number of semaphores systemwide • Maximum number of semaphore operations that can be performed per semaphore system call • Maximum number of semaphore sets systemwide
`net.ipv4.ip_local_port_range = 9000 65500`	Range of port numbers
`net.core.rmem_default = 262144`	Default OS TCP receive buffer size for all types of connections
`net.core.rmem_max = 16777216`	Max OS TCP receive buffer size for all types of connections
`net.core.wmem_default = 262144`	Default OS TCP send buffer size for all types of connections
`net.core.wmem_max = 16777216`	Max OS TCP send buffer size for all types of connections

A complete listing of our `/etc/sysctl.conf` file can be found in section 8.1 of this document. Out of all the parameter settings applied, the greatest performance improvement was achieved by increasing the `sunrpc.tcp_slot_table_entries` parameter to `128`. This setting gave us a throughput increase of about 6%.

6.3 NETWORKING

We applied the following network configurations in our test environment:

- **Jumbo frames (9,000 MTU).** The use of jumbo frames is a way to achieve higher throughput and better CPU utilization. Jumbo frames are particularly useful for database transfers when used with 10GbE and sequential read-oriented workloads.
- **Host-side 10GbE NIC tuning.** A number of parameters can be set for the 10GbE driver. We ran tests with several different modified parameter settings (one at a time). The only one that made any noticeable difference in performance with our workload was the RSS parameter. The RSS parameter is used to influence receive side scaling. Receive side scaling allows the receive side network load to be spread across multiple CPUs. A setting of "0" disables it, while the default value of "1" both enables it and sets the queue count to either 16 or the number of online CPUs, whichever is less. This parameter can be set as high as 16. With our CPU-intensive workload, we found that a setting of 6 gave us a throughput increase of about 2% on average. This parameter can be set by using syntax similar to `modprobe ixgbe RSS=6` or by appending the following line to the `/etc/modprobe.conf` file and rerunning `modprobe` or rebooting: `options ixgbe RSS=6.`
- **Dedicated VLAN per I/O (10GbE) network.** To provide security and traffic isolation between subnets.

6.4 NETAPP STORAGE SYSTEM

The following lists the volume-level options configured on the NetApp unified storage controllers for all of the test configurations:

- Fibre Channel loops connecting controllers to shelves set to run at 4Gb/s.
- Jumbo frames enabled (MTU size of 9,000) on all data network interfaces. This is especially critical for 10GbE networks. A failure to do this can result in performance degradation as high as 40%.
- WAFL® (Write Anywhere File Layout) volume options used for data file volumes:
  - `max_write_alloc_blocks=256`. Setting `max_write_alloc_blocks` to `256` optimizes the WAFL on-disk data layout for large sequential workloads and is strongly recommended for large block sequential database workloads. It should be set before any data is written to the volume.
  - `no_atime_update=on`. Enabling `no_atime_update` reduces I/O overhead by avoiding updates to the access time attribute each time a file is accessed. The impact of enabling this flag is completely workload related. This setting made no noticeable change in performance with our large block sequential workload; however, it is always advisable with Oracle databases.
- Data ONTAP TCP receive and transfer buffer settings:
  - `nfs.tcp.recvwindowsize = 65536`
  - `nfs.tcp.xfersize = 65536`

6.5 VOLUME MOUNT OPTIONS

We used the following mount options for our data file and log volumes, per Oracle RAC best practices and requirements:

`rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp`

Refer to NetApp KB ID 3010189 for a more detailed listing of NFS mount option best practices for Oracle databases.

6.6 ORACLE RAC/RDBMS

Multiple areas under Oracle can be optimized for performance. We used the following guidelines when setting up our test environment:

- **Tablespaces.** Key tablespaces were striped across all controllers (by distributing data files of each tablespace across the controllers).
- **Partitions.** Partitioning of large tables and indexes was employed to enhance overall scalability of the RAC and to improve the performance on SQL operations such as index range scans, full table scans, joins, and updates. Partitioning is key for optimizing the way the Oracle server performs parallel execution.

  Operations on partitioned tables and indexes are performed in parallel by assigning different parallel execution servers to different partitions of the table or index. In addition, partitioning enables the mapping of partitions to volumes and controllers for better I/O load balancing.
- **Parallel server and execution.** In addition to partitioned tables and indexes, using parallel execution can improve performance in data warehouses. A few Oracle `init` parameters related to parallel server tuning were set in our test environment:
  - `parallel_execution_message_size=16384` (specifies the size of message buffers used for communication by parallel processes)
  - `parallel_force_local=TRUE` (restricts parallel server processes to the node where the query is issued). This setting is not necessarily a best practice, but it enabled us to use a building block approach to the design of our modular scalable implementation.
- Others:
  - `db_block_size=65536` (to set the database block size to 64KB)

- db_file_multiblock_read_count=32 (reduced from the default of 128 to minimize excessive I/O caused by read-ahead operations when using a large database block size)
- filesystemio_options='setall' (to enable direction and async I/O)

In addition, the following parameters were configured with appropriate values to suit our workload:

```
pga_aggregate_target=4294967296
processes=500
open_cursors=300
sessions=555
sga_target=8589934592
resource_manager_plan='FORCE:INTERNAL_PLAN'
```

We used `resource_manager_plan='FORCE:INTERNAL_PLAN'` to force equal allocation of computing resources to all users and to provide consistent test results.

7 SUMMARY

Through the careful execution of a comprehensive test plan, we demonstrated the modular scalable architecture of Oracle 11*g* RAC databases capable of supporting the large block sequential I/O requirements of modern data warehouses. We used the Oracle Direct NFS client, the latest Ethernet technology (10GbE), NetApp unified storage, and workloads typically used with large-scale data warehouses. In addition, we demonstrated the improved performance and scaling capabilities of DNFS over the Linux kernel NFS client. A summary of our solution is as follows:

- Using the configuration and tools described in this paper, Oracle 11*g* RAC databases on NetApp storage can scale in a near-linear fashion.
- Through the more efficient use of client CPU resources, Oracle's new DNFS client can significantly outperform the traditional Linux kernel NFS client.
- NetApp unified storage provides the scalability, high performance, and manageability required to support large-scale data warehouse deployments using 10GbE networks and standard Ethernet protocols.

The continued evolution of Ethernet technology has resulted in the availability of 10GbE, with 40GbE and 100GbE in development. These developments have radically changed the way data is stored and transported in the modern data center, breaking through the old barriers that previously limited the size, scalability, and manageability of large-scale Oracle data warehouses. The combination of Oracle Real Application Clusters, Oracle DNFS, 10GbE, and NetApp unified storage has enabled the design and deployment of a modular scalable architecture that is capable of supporting large-scale databases.

8 APPENDIXES

8.1 SERVER AND OPERATING SYSTEM CONFIGURATIONS

OPERATING SYSTEM VERSION

```
RHEL 5.3
Linux version 2.6.18-128.el5 (mockbuild@hs20-bc1-7.build.redhat.com) (gcc
version 4.1.2 20080704 (Red Hat 4.1.2-44)) #1 SMP
```

10GBE NETWORK INTERFACE DRIVER VERSION

```
ethtool -i eth3:
driver: ixgbe
```

```
version: 3.0.12-NAPI
firmware-version: 0.9-3
bus-info: 0000:07:00.1
```

10GBE NETWORK INTERFACE HOST-SIDE OPTIMIZATION (/ETC/MODPOBE.CONF)

```
options ixgbe RSS=6
```

LINUX KERNEL SETTINGS

```
/etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename
# Useful for debugging multi-threaded applications
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Controls the maximum size of a message, in bytes
kernel.msgmnb = 65536

# Controls the default maxmimum size of a mesage queue
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 21474836480


# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

# Additional settings for Oracle 11gR2
fs.file-max = 6815744
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.aio-max-nr = 1048576

# Additional tunings for NFS etc
sunrpc.tcp_slot_table_entries = 128
net.core.rmem_default = 262144
```

```
net.core.rmem_max = 16777216
net.core.wmem_default = 262144
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 262144 16777216
net.ipv4.tcp_wmem = 4096 262144 16777216
net.ipv4.tcp_syncookies = 0
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 0
net.ipv4.tcp_window_scaling = 1
```

NETWORK INTERFACE CONFIGURATIONS

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:1A:64:D2:D8:7C
          inet addr:10.61.172.38  Bcast:10.61.172.255  Mask:255.255.255.0
          inet6 addr: fe80::21a:64ff:fed2:d87c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:311454 errors:0 dropped:0 overruns:0 frame:0
          TX packets:205904 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:70494758 (67.2 MiB)  TX bytes:73635570 (70.2 MiB)
          Interrupt:106 Memory:ce000000-ce012100

eth0:1    Link encap:Ethernet  HWaddr 00:1A:64:D2:D8:7C
          inet addr:10.61.172.91  Bcast:10.61.172.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:106 Memory:ce000000-ce012100

eth1      Link encap:Ethernet  HWaddr 00:1B:21:5A:6A:70
          inet addr:192.168.3.102  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::21b:21ff:fe5a:6a70/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:360828622 errors:0 dropped:961710 overruns:0 frame:0
          TX packets:277762850 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6504285597151 (5.9 TiB)  TX bytes:30633325163 (28.5 GiB)

eth2      Link encap:Ethernet  HWaddr 00:1A:64:D2:D8:7E
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::21a:64ff:fed2:d87e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10640039 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13355714 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8213603729 (7.6 GiB)  TX bytes:12204868543 (11.3 GiB)
          Interrupt:169 Memory:ca000000-ca012100

eth3      Link encap:Ethernet  HWaddr 00:1B:21:5A:6A:71
          inet addr:192.168.2.102  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::21b:21ff:fe5a:6a71/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:375348399 errors:0 dropped:1202965 overruns:0 frame:0
          TX packets:290741122 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6740082824200 (6.1 TiB)  TX bytes:33145505319 (30.8 GiB)

lo        Link encap:Local Loopback
```

```
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:762786 errors:0 dropped:0 overruns:0 frame:0
          TX packets:762786 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:257186580 (245.2 MiB)  TX bytes:257186580 (245.2 MiB)

sit0      Link encap:IPv6-in-IPv4
          NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

HOSTS FILE

```
/etc/hosts
127.0.0.1       refex-1.rtp.netapp.com refex-1 localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
10.61.172.38    teso-x3650-1 teso-x3650-1.rtp.netapp.com
192.168.3.102   teso-x3650-1-eth1
192.168.2.102   teso-x3650-1-eth3
10.61.172.40    teso-x3650-2 teso-x3650-2.rtp.netapp.com
192.168.3.105   teso-x3650-2-eth1
192.168.2.105   teso-x3650-2-eth3
10.61.172.34    teso-x3650-3 teso-x3650-3.rtp.netapp.com
192.168.3.108   teso-x3650-3-eth1
192.168.2.108   teso-x3650-3-eth3
10.61.172.44    teso-x3650-4 teso-x3650-4.rtp.netapp.com
192.168.3.111   teso-x3650-4-eth1
192.168.2.111   teso-x3650-4-eth3
10.61.172.46    teso-x3650-5 teso-x3650-5.rtp.netapp.com
192.168.3.114   teso-x3650-5-eth1
192.168.2.114   teso-x3650-5-eth3
10.61.172.48    teso-x3650-6 teso-x3650-6.rtp.netapp.com
192.168.3.117   teso-x3650-6-eth1
192.168.2.117   teso-x3650-6-eth3
10.61.172.70    teso-x3650-7 teso-x3650-7.rtp.netapp.com
192.168.3.120   teso-x3650-7-eth1
192.168.2.120   teso-x3650-7-eth3
10.61.172.72    teso-x3650-8 teso-x3650-8.rtp.netapp.com
192.168.3.123   teso-x3650-8-eth1
192.168.2.123   teso-x3650-8-eth3
10.61.172.135   teso-fas3170-9
192.168.2.135   teso-fas3170-9-e1a
192.168.3.145   teso-fas3170-9-e1b
192.168.4.135   teso-fas3170-9-vif1
10.61.172.136   teso-fas3170-10
192.168.2.136   teso-fas3170-10-e1a
192.168.3.146   teso-fas3170-10-e1b
192.168.4.136   teso-fas3170-10-vif1
10.61.172.137   teso-fas3170-11
192.168.2.137   teso-fas3170-11-e1a
192.168.3.147   teso-fas3170-11-e1b
192.168.4.137   teso-fas3170-11-vif1
10.61.172.138   teso-fas3170-12
192.168.2.138   teso-fas3170-12-e1a
192.168.3.148   teso-fas3170-12-e1b
192.168.4.138   teso-fas3170-12-vif1
10.61.172.139   teso-fas3170-13
```

```
192.168.2.139   teso-fas3170-13-e1a
192.168.3.149   teso-fas3170-13-e1b
192.168.4.139   teso-fas3170-13-vif1
10.61.172.140   teso-fas3170-14
192.168.2.140   teso-fas3170-14-e1a
192.168.3.150   teso-fas3170-14-e1b
192.168.4.140   teso-fas3170-14-vif1
10.61.172.141   teso-fas3170-15
192.168.2.141   teso-fas3170-15-e1a
192.168.3.151   teso-fas3170-15-e1b
192.168.4.141   teso-fas3170-15-vif1
10.61.172.142   teso-fas3170-16
192.168.2.142   teso-fas3170-16-e1a
192.168.3.152   teso-fas3170-16-e1b
192.168.4.142   teso-fas3170-16-vif1

# Oracle RAC VIPs
10.61.172.91    teso-x3650-1-vip
10.61.172.92    teso-x3650-2-vip
10.61.172.93    teso-x3650-3-vip
10.61.172.94    teso-x3650-4-vip
10.61.172.95    teso-x3650-5-vip
10.61.172.96    teso-x3650-6-vip
10.61.172.97    teso-x3650-7-vip
10.61.172.98    teso-x3650-8-vip

# Oracle RAC Cluster Interconnects
192.168.1.101   teso-x3650-1-eth2 teso-x3650-1-priv
192.168.1.104   teso-x3650-2-eth2 teso-x3650-2-priv
192.168.1.107   teso-x3650-3-eth2 teso-x3650-3-priv
192.168.1.110   teso-x3650-4-eth2 teso-x3650-4-priv
192.168.1.113   teso-x3650-5-eth2 teso-x3650-5-priv
192.168.1.116   teso-x3650-6-eth2 teso-x3650-6-priv
192.168.1.119   teso-x3650-7-eth2 teso-x3650-7-priv
192.168.1.122   teso-x3650-8-eth2 teso-x3650-8-priv

# Oracle Grid SCAN
10.61.172.99    dnfs-cluster-scan
```

VOLUMES AND MOUNT OPTIONS

```
/etc/vfstab
/dev/VolGroup00/LogVol00 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
/dev/VolGroup00/LogVol01 swap                    swap    defaults        0 0
/dev/hdc                /mnt/dvd                auto    noauto,users    0 0
teso-fas3170-9-e1a:/vol/vol_data1  /oradata1 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-10-e1a:/vol/vol_data1  /oradata2 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-11-e1a:/vol/vol_data1  /oradata3 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-12-e1a:/vol/vol_data1  /oradata4 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-13-e1a:/vol/vol_data1  /oradata5 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-14-e1a:/vol/vol_data1  /oradata6 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
```

```
teso-fas3170-15-e1a:/vol/vol_data1  /oradata7 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-16-e1a:/vol/vol_data1  /oradata8 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-9-e1b:/vol/vol_data2  /oradata9 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-10-e1b:/vol/vol_data2  /oradata10 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-11-e1b:/vol/vol_data2  /oradata11 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-12-e1b:/vol/vol_data2  /oradata12 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-13-e1b:/vol/vol_data2  /oradata13 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-14-e1b:/vol/vol_data2  /oradata14 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-15-e1b:/vol/vol_data2  /oradata15 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-16-e1b:/vol/vol_data2  /oradata16 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-9-e1a:/vol/vol_logs  /oralog1 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-10-e1b:/vol/vol_logs  /oralog2 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-11-e1a:/vol/vol_logs  /oralog3 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-12-e1b:/vol/vol_logs  /oralog4 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-13-e1a:/vol/vol_logs  /oralog5 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-14-e1b:/vol/vol_logs  /oralog6 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-15-e1a:/vol/vol_logs  /oralog7 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-16-e1b:/vol/vol_logs  /oralog8 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-10-e1a:/vol/vol_ocr  /oraocr1 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-12-e1a:/vol/vol_ocr  /oraocr2 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-14-e1a:/vol/vol_ocr  /oraocr3 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-15-e1a:/vol/vol_extdata  /oraextdata1 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
teso-fas3170-16-e1b:/vol/vol_extdata  /oraextdata2 nfs
rw,bg,hard,rsize=65536,wsize=65536,vers=3,actimeo=0,nointr,suid,timeo=600,tcp
```

8.2 ORACLE CONFIGURATIONS

SPFILE FILE

```
*.audit_file_dest='/oracle/app/admin/TPCH/adump'
*.audit_trail='db'
*.cluster_database=true
*.compatible='11.2.0.0.0'
*.control_files='/oralog7/oradata/TPCH/controlfile/o1_mf_61fg0dw7_.ctl','/ora
log8/oradata/TPCH/controlfile/o1_mf_61fg0f48_.ctl'
*.db_block_size=32768
*.db_create_file_dest='/oradCCgata1/oradata'
*.db_create_online_log_dest_1='/oralog7/oradata'
```

```
*.db_create_online_log_dest_2='/oralog8/oradata'
*.db_domain=''
*.db_file_multiblock_read_count=32
*.db_files=400
*.db_name='TPCH'
*.diagnostic_dest='/oracle/app'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TPCHXDB)'
*.filesystemio_options='setall'
TPCH5.instance_number=5
TPCH7.instance_number=7
TPCH4.instance_number=4
TPCH6.instance_number=6
TPCH1.instance_number=1
TPCH2.instance_number=2
TPCH3.instance_number=3
TPCH8.instance_number=8
*.nls_date_format='YYYY-MM-DD'
*.open_cursors=300
*.parallel_degree_policy='LIMITED'
*.parallel_execution_message_size=16384
*.parallel_force_local=TRUE
*.parallel_max_servers=64
*.parallel_server_instances=8
*.parallel_servers_target=32
*.parallel_threads_per_cpu=2
*.pga_aggregate_target=4294967296
*.processes=500
*.remote_listener='dnfs-cluster-scan:1521'
*.remote_login_passwordfile='exclusive'
*.resource_manager_plan='FORCE:INTERNAL_PLAN'
*.sessions=555
*.sga_max_size=1589934592
*.sga_target=8589934592
TPCH8.thread=8
TPCH6.thread=6
TPCH5.thread=5
TPCH4.thread=4
TPCH3.thread=3
TPCH2.thread=2
TPCH7.thread=7
TPCH1.thread=1
TPCH8.undo_tablespace='UNDOTBS8'
TPCH5.undo_tablespace='UNDOTBS5'
TPCH3.undo_tablespace='UNDOTBS3'
TPCH4.undo_tablespace='UNDOTBS4'
TPCH1.undo_tablespace='UNDOTBS1'
TPCH7.undo_tablespace='UNDOTBS7'
TPCH2.undo_tablespace='UNDOTBS2'
TPCH6.undo_tablespace='UNDOTBS6'
```

DNFS CONFIGURATIONS

```
$ORACLE_HOME/dbs/oranfstab
server: teso-fas3170-9
local: teso-x3650-1-eth3 path: teso-fas3170-9-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-9-e1b
export:/vol/vol_data1 mount:/oradata1
```

```
export:/vol/vol_data2 mount:/oradata9
export:/vol/vol_logs mount:/oralog1

server: teso-fas3170-10
local: teso-x3650-1-eth3 path: teso-fas3170-10-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-10-e1b
export:/vol/vol_data1 mount:/oradata2
export:/vol/vol_data2 mount:/oradata10
export:/vol/vol_logs mount:/oralog2

server: teso-fas3170-11
local: teso-x3650-1-eth3 path: teso-fas3170-11-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-11-e1b
export:/vol/vol_data1 mount:/oradata3
export:/vol/vol_data2 mount:/oradata11
export:/vol/vol_logs mount:/oralog3

server: teso-fas3170-12
local: teso-x3650-1-eth3 path: teso-fas3170-12-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-12-e1b
export:/vol/vol_data1 mount:/oradata4
export:/vol/vol_data2 mount:/oradata12
export:/vol/vol_logs mount:/oralog4

server: teso-fas3170-13
local: teso-x3650-1-eth3 path: teso-fas3170-13-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-13-e1b
export:/vol/vol_data1 mount:/oradata5
export:/vol/vol_data2 mount:/oradata13
export:/vol/vol_logs mount:/oralog5

server: teso-fas3170-14
local: teso-x3650-1-eth3 path: teso-fas3170-14-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-14-e1b
export:/vol/vol_data1 mount:/oradata6
export:/vol/vol_data2 mount:/oradata14
export:/vol/vol_logs mount:/oralog6

server: teso-fas3170-15
local: teso-x3650-1-eth3 path: teso-fas3170-15-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-15-e1b
export:/vol/vol_data1 mount:/oradata7
export:/vol/vol_data2 mount:/oradata15
export:/vol/vol_logs mount:/oralog7

server: teso-fas3170-16
local: teso-x3650-1-eth3 path: teso-fas3170-16-e1a
local: teso-x3650-1-eth1 path: teso-fas3170-16-e1b
export:/vol/vol_data1 mount:/oradata8
export:/vol/vol_data2 mount:/oradata16
export:/vol/vol_logs mount:/oralog8
```

CLUSTERWARE AND INSTANCE CONFIGURATIONS

crs_stat outpout

Name	Type	Target	State	Host
ora....ER.lsnr	ora....er.type	ONLINE	ONLINE	teso...50-1
ora....N1.lsnr	ora....er.type	ONLINE	ONLINE	teso...50-5
ora.asm	ora.asm.type	OFFLINE	OFFLINE	
ora.eons	ora.eons.type	ONLINE	ONLINE	teso...50-1
ora.gsd	ora.gsd.type	OFFLINE	OFFLINE	

```
ora....network ora....rk.type ONLINE    ONLINE    teso...50-1
ora.oc4j       ora.oc4j.type  OFFLINE   OFFLINE
ora.ons        ora.ons.type   ONLINE    ONLINE    teso...50-1
ora....ry.acfs ora....fs.type OFFLINE   OFFLINE
ora.scan1.vip  ora....ip.type ONLINE    ONLINE    teso...50-5
ora....SM1.asm application    OFFLINE   OFFLINE
ora....-1.lsnr application    ONLINE    ONLINE    teso...50-1
ora....0-1.gsd application    OFFLINE   OFFLINE
ora....0-1.ons application    ONLINE    ONLINE    teso...50-1
ora....0-1.vip ora....t1.type ONLINE    ONLINE    teso...50-1
ora....SM2.asm application    OFFLINE   OFFLINE
ora....-2.lsnr application    ONLINE    ONLINE    teso...50-2
ora....0-2.gsd application    OFFLINE   OFFLINE
ora....0-2.ons application    ONLINE    ONLINE    teso...50-2
ora....0-2.vip ora....t1.type ONLINE    ONLINE    teso...50-2
ora....SM3.asm application    OFFLINE   OFFLINE
ora....-3.lsnr application    ONLINE    ONLINE    teso...50-3
ora....0-3.gsd application    OFFLINE   OFFLINE
ora....0-3.ons application    ONLINE    ONLINE    teso...50-3
ora....0-3.vip ora....t1.type ONLINE    ONLINE    teso...50-3
ora....SM4.asm application    OFFLINE   OFFLINE
ora....-4.lsnr application    ONLINE    ONLINE    teso...50-4
ora....0-4.gsd application    OFFLINE   OFFLINE
ora....0-4.ons application    ONLINE    ONLINE    teso...50-4
ora....0-4.vip ora....t1.type ONLINE    ONLINE    teso...50-4
ora....SM5.asm application    OFFLINE   OFFLINE
ora....-5.lsnr application    ONLINE    ONLINE    teso...50-5
ora....0-5.gsd application    OFFLINE   OFFLINE
ora....0-5.ons application    ONLINE    ONLINE    teso...50-5
ora....0-5.vip ora....t1.type ONLINE    ONLINE    teso...50-5
ora....SM6.asm application    OFFLINE   OFFLINE
ora....-6.lsnr application    ONLINE    ONLINE    teso...50-6
ora....0-6.gsd application    OFFLINE   OFFLINE
ora....0-6.ons application    ONLINE    ONLINE    teso...50-6
ora....0-6.vip ora....t1.type ONLINE    ONLINE    teso...50-6
ora....SM7.asm application    OFFLINE   OFFLINE
ora....-7.lsnr application    ONLINE    ONLINE    teso...50-7
ora....0-7.gsd application    OFFLINE   OFFLINE
ora....0-7.ons application    ONLINE    ONLINE    teso...50-7
ora....0-7.vip ora....t1.type ONLINE    ONLINE    teso...50-7
ora....SM8.asm application    OFFLINE   OFFLINE
ora....-8.lsnr application    ONLINE    ONLINE    teso...50-8
ora....0-8.gsd application    OFFLINE   OFFLINE
ora....0-8.ons application    ONLINE    ONLINE    teso...50-8
ora....0-8.vip ora....t1.type ONLINE    ONLINE    teso...50-8
ora.tpch.db    ora....se.type ONLINE    ONLINE    teso...50-1
```

8.3 NETAPP STORAGE SYSTEM CONFIGURATIONS

DATA ONTAP VERSION

```
NetApp Release 8.0 7-Mode: Thu Mar 11 16:17:13 PST 2010
```

VOLUME OPTIONS (EXAMPLES SHOWING VOLUME OPTIONS FROM ONE CONTROLLER)

```
teso-fas3170-9> vol options /vol/vol_logs
```

```
nosnap=on, nosnapdir=off, minra=off, no_atime_update=on, nvfail=off,
ignore_inconsistent=off, snapmirrored=off, create_ucode=off,
convert_ucode=off, maxdirsize=167690, schedsnapname=ordinal,
fs_size_fixed=off, compression=off, guarantee=none, svo_enable=off,
svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
no_i2p=off, fractional_reserve=100, max_write_alloc_blocks=256, extent=off,
try_first=volume_grow, read_realloc=off, snapshot_clone_dependency=off,
nbu_archival_snap=off

teso-fas3170-9> vol options /vol/vol_ocr
nosnap=on, nosnapdir=off, minra=off, no_atime_update=on, nvfail=off,
ignore_inconsistent=off, snapmirrored=off, create_ucode=off,
convert_ucode=off, maxdirsize=167690, schedsnapname=ordinal,
fs_size_fixed=off, compression=off, guarantee=none, svo_enable=off,
svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
no_i2p=off, fractional_reserve=100, max_write_alloc_blocks=256, extent=off,
try_first=volume_grow, read_realloc=off, snapshot_clone_dependency=off,
nbu_archival_snap=off

teso-fas3170-9> vol options /vol/vol_data1
nosnap=on, nosnapdir=off, minra=off, no_atime_update=on, nvfail=off,
ignore_inconsistent=off, snapmirrored=off, create_ucode=off,
convert_ucode=off, maxdirsize=167690, schedsnapname=ordinal,
fs_size_fixed=off, compression=off, guarantee=none, svo_enable=off,
svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
no_i2p=off, fractional_reserve=100, max_write_alloc_blocks=256, extent=off,
try_first=volume_grow, read_realloc=off, snapshot_clone_dependency=off,
nbu_archival_snap=off

teso-fas3170-9> vol options /vol/vol_data2
nosnap=on, nosnapdir=off, minra=off, no_atime_update=on, nvfail=off,
ignore_inconsistent=off, snapmirrored=off, create_ucode=off,
convert_ucode=off, maxdirsize=167690, schedsnapname=ordinal,
fs_size_fixed=off, compression=off, guarantee=none, svo_enable=off,
svo_checksum=off, svo_allow_rman=off, svo_reject_errors=off,
no_i2p=off, fractional_reserve=100, max_write_alloc_blocks=256, extent=off,
try_first=volume_grow, read_realloc=off, snapshot_clone_dependency=off,
nbu_archival_snap=off
```

NETWORK INTERFACE CONFIGURATIONS

```
e0M: flags=0x2b0c866<BROADCAST,RUNNING,MULTICAST,TCPCKSUM> mtu 1500
        ether 00:a0:98:12:e1:2a (auto-100tx-fd-up) flowcontrol full
e0a: flags=0x2f4c867<UP,BROADCAST,RUNNING,MULTICAST,TCPCKSUM> mtu 1500
        inet 10.61.172.135 netmask 0xffffff00 broadcast 10.61.172.255
        ether 00:a0:98:12:e1:28 (auto-1000t-fd-up) flowcontrol full
e0b: flags=0x270c866<BROADCAST,RUNNING,MULTICAST,TCPCKSUM> mtu 1500
        ether 00:a0:98:12:e1:29 (auto-unknown-down) flowcontrol full
e1a: flags=0x5f4c867<UP,BROADCAST,RUNNING,MULTICAST,TCPCKSUM,NOWINS> mtu 9000
        inet 192.168.2.135 netmask 0xffffff00 broadcast 192.168.2.255
        ether 00:07:43:05:68:93 (auto-10g_sr-fd-up) flowcontrol full
e1b: flags=0x5f4c867<UP,BROADCAST,RUNNING,MULTICAST,TCPCKSUM,NOWINS> mtu 9000
        inet 192.168.3.145 netmask 0xffffff00 broadcast 192.168.3.255
        ether 00:07:43:05:68:94 (auto-10g_sr-fd-up) flowcontrol full
lo: flags=0x1b48049<UP,LOOPBACK,RUNNING,MULTICAST,TCPCKSUM> mtu 8160
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.0.0.1
        ether 00:00:00:00:00:00 (VIA Provider)
```

```
losk: flags=0x40a400c9<UP,LOOPBACK,RUNNING> mtu 9188
        inet 127.0.20.1 netmask 0xff000000 broadcast 127.0.20.1
```

9 REFERENCES

- Oracle Real Application Clusters (RAC) 11*g* Release 2: An Oracle White Paper
  www.oracle.com/technetwork/database/clustering/overview/twp-rac11gr2-134105.pdf
- Oracle Database 11*g* Direct NFS Client: An Oracle White Paper
  www.oracle.com/technetwork/articles/directnfsclient-11gr1-twp-129785.pdf
- 10 Gigabit Ethernet Technology Overview and Applications for Enterprise Data Centers
  www.bladenetwork.net/userfiles/file/PDFs/WP_10GbE_Tech_Overview.pdf
- Oracle Database Installation Guide 11*g* Release 2 (11.2) for Linux
  http://download.oracle.com/docs/cd/E11882_01/install.112/e16763/pre_install.htm#CIHEGJEH

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

www.netapp.com