ORACLE 10*g*™ PERFORMANCE – PROTOCOL COMPARISON ON SUN™ SOLARIS™ 10

A Comparison of Oracle 10*g* Performance with NFSv3, FCP, and iSCSI

John Elliott | Network Appliance, Inc.

September 2006 | TR-3496

TABLE OF CONTENTS

TR-3496 Oracle 10*g*™ Performance – Protocol Comparison on Sun™ Solaris™ 10 1
Introduction 3
Section 1: Executive Summary 3
Test Configurations 3
Database Workload Description 4
OLTP Database Load Results Summary... 4
Figure 1) Overall database throughput comparison for all storage protocols tested. 5
Figure 2) Host CPU utilization for database throughput tests. 6
Raw I/O Throughput Test Description and Results 6
Figure 3) Raw I/O throughput for database configurations tested. 7
Full Table Scan Tests with Oracle Parallel Query – Description and Results......................... 7
Figure 4) Full Table Scans – impact of increasing db_file_multiblock_read_count and invoking Oracle Parallel Query (Degree Of Parallelism of "1" indicates Parallel Query was not enabled). 8
Section 2: Details - FCP Environment 9
Database Layout 9
Storage Network Configuration.. 9
Figure 5) FCP network diagram. 9
Section 3: Details of Hardware iSCSI Environment 10
Database Layout 10
Storage Network Configuration.. 10
Figure 6) Hardware iSCSI network diagram. 10
Section 4: Details of Software iSCSI Environment 11
Database Layout 11
Storage Network Configuration.. 11
Figure 7) Software iSCSI and NFSv3 network diagram. 11
Section 5: Details of NFSv3 Environment.. 12
Database Layout 12
Storage Network Configuration.. 12
Section 6: Conclusion.. 12
Appendix A: Oracle Parallel Query – Syntax and Application 13
Appendix B: Pertinent Solaris Kernel Parameter Settings with Syntax.................................. 13
Appendix C: Pertinent NetApp Storage System Settings 13
Appendix D: Mount Options.. 13
Appendix E: Patches, Drivers, and Software.. 13
Appendix F: Pertinent Oracle Parameter (pfile) Settings 14
OLTP Throughput Tests 14
Parallel Query Tests.. 14
Appendix G: Lessons Learned, Bugs, etc. 15
Oracle Parameter "_kgl_large_heap_warning_threshold"... 15
Solaris 10 UFS LUN Misalignment Problem.. 15
Appendix H: Pertinent Hardware Details 16
Appendix I: Adherence To Best Practices 16
Acknowledgements... 16

Introduction

Network Appliance, Inc., provides the ideal storage platform for enterprise Oracle® databases requiring high availability and high performance. NetApp storage not only supports but excels at protecting and serving data with all the major storage protocols, including NFSv3, FCP, and iSCSI. In support of enterprise data systems, NetApp storage systems provide superior, cost-effective data protection, backup and recovery, availability, and administration through Network Appliance™ tools and features that include the following:

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

The purpose of this paper is to present test results that will clearly demonstrate the high performance of NetApp storage with Oracle databases running on Solaris 10. To support these results, several points of comparison are used, including the following:

- Oracle 10g database throughput measurements from several of the major storage protocols
    - NFSv3 (with and without jumbo frames)
    - FCP
    - iSCSI with software initiators
    - iSCSI with hardware initiators and targets
- Raw I/O throughput data using a NetApp load generator with the following storage protocols:
    - NFSv3
    - FCP
    - iSCSI with software initiators
    - iSCSI with hardware initiators and targets

There is also a discussion of Oracle Parallel Query along with test data demonstrating the performance advantages of using Parallel Query in cases where full table scans are unavoidable.

Section 1: Executive Summary

Test Configurations

Our test bed consisted of a Fujitsu PrimePower 650 8-way server and two NetApp storage systems. The software environment included Solaris 10 and Oracle 10.2.0.1.0 on the Fujitsu server and Data ONTAP® 7.1RC4 on the storage system. We tested the following storage configurations:

- NFSv3 (with and without jumbo frames) using Intel® Pro/1000 NICs
- FCP using QLogic™ QLA 2342 adapters
- iSCSI with software initiators using Intel Pro/1000 NICs
- iSCSI with hardware initiators and targets using QLogic QLA 4010 HBAs and Emulex™ LP100i-D1 target cards

FCP and iSCSI LUNs were configured and mounted as UFS file systems for database storage.

It’s very important to understand that we designed our test environments to stress the database server software and hardware with the goal of completely utilizing all database server CPU resources for each round of tests. To this end, we avoided resource bottlenecks in other areas, including physical memory, storage capacity and bandwidth, and network bandwidth through our choice of server hardware, network configuration, and the two NetApp storage systems for storing the database files.

Additional details of the hardware and network configuration can be found in Appendix H.

Database Workload Description

The database used for testing can best be described as OLTP (Online Transaction Processing) in nature with a physical size of approximately 420 gigabytes. During the testing, we used a set of scripts and executables to generate an OLTP-type load consisting of a steady stream of small, random read and write operations (approximately 57% reads and 43% writes) against the test database. This workload was designed to emulate the real-life activities of a wholesale supplier's order processing system in which inventory is spread across several regional warehouses. Within that framework, a single order consisted of multiple database transactions with orders averaging 10 items each. In terms of actual database transactions, each item ordered resulted in all of the following database transactions:

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

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

Each round of tests consisted of the following:

- 2 test cycles of 12 minutes each
- 2 test cycles of 68 minutes each

The data presented in this paper was recorded from the last 68-minute cycle of each round.

OLTP Database Load Results Summary

As expected, FCP was the highest performer in terms of throughput. The results are listed below:

- FCP with UFS – 39625 Order Entry Transactions Per Minute
- Hardware iSCSI with UFS – 35660 Order Entry Transactions Per Minute
- NFSv3 with Jumbo Frames – 31408 Order Entry Transactions Per Minute
- Software iSCSI with UFS – 28513 Order Entry Transactions Per Minute
- NFSv3 without Jumbo Frames – 25134 Order Entry Transactions Per Minute

See Figure 1 below for a graphical summary of these test results:

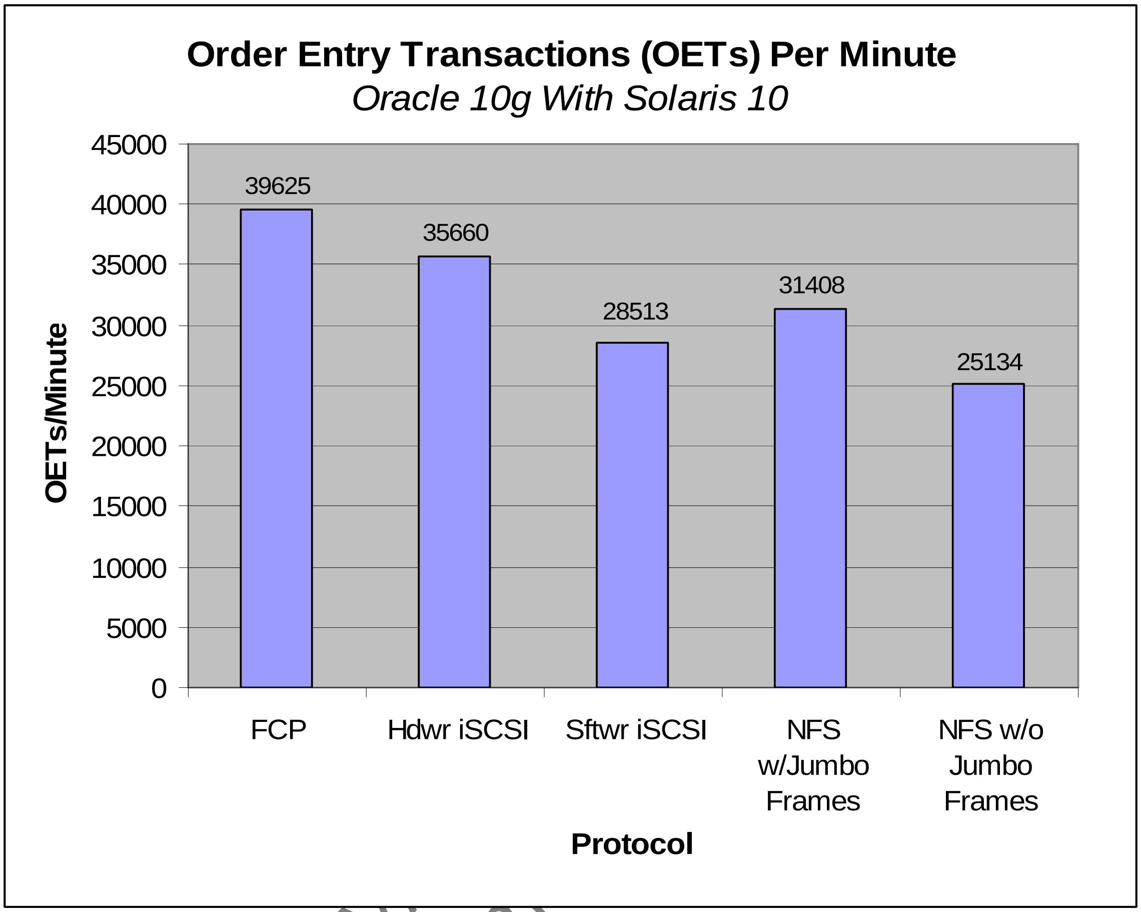

Figure 1) Overall database throughput comparison for all storage protocols tested.

Using FCP throughput as a point of comparison, the results can be summarized as follows:

- Throughput of hardware iSCSI with UFS was 10% lower than that of FCP with UFS.
- Throughput of NFSv3 with jumbo frames was 21% lower than that of FCP (with UFS).
- Throughput of software iSCSI with jumbo frames was 29% lower than that of FCP.
- Throughput of NFSv3 without jumbo frames was 37% lower than that of FCP.

Another point of interest is database host CPU utilization. The UNIX® “mpstat” tool reports CPU utilization in terms of “user,” “system,” “wait,” and “idle” time. While we were never able to drive CPU idle time all the way down to “0” we were able to consistently drive CPU wait time to “0.” In addition to that we observed that CPU “user” time basically followed the same trend of database throughput per storage protocol. For example, user CPU time for the FCP environment was highest while user time for NFSv3 without jumbo frames was lowest. See Figure 2 for a summary of CPU utilization.

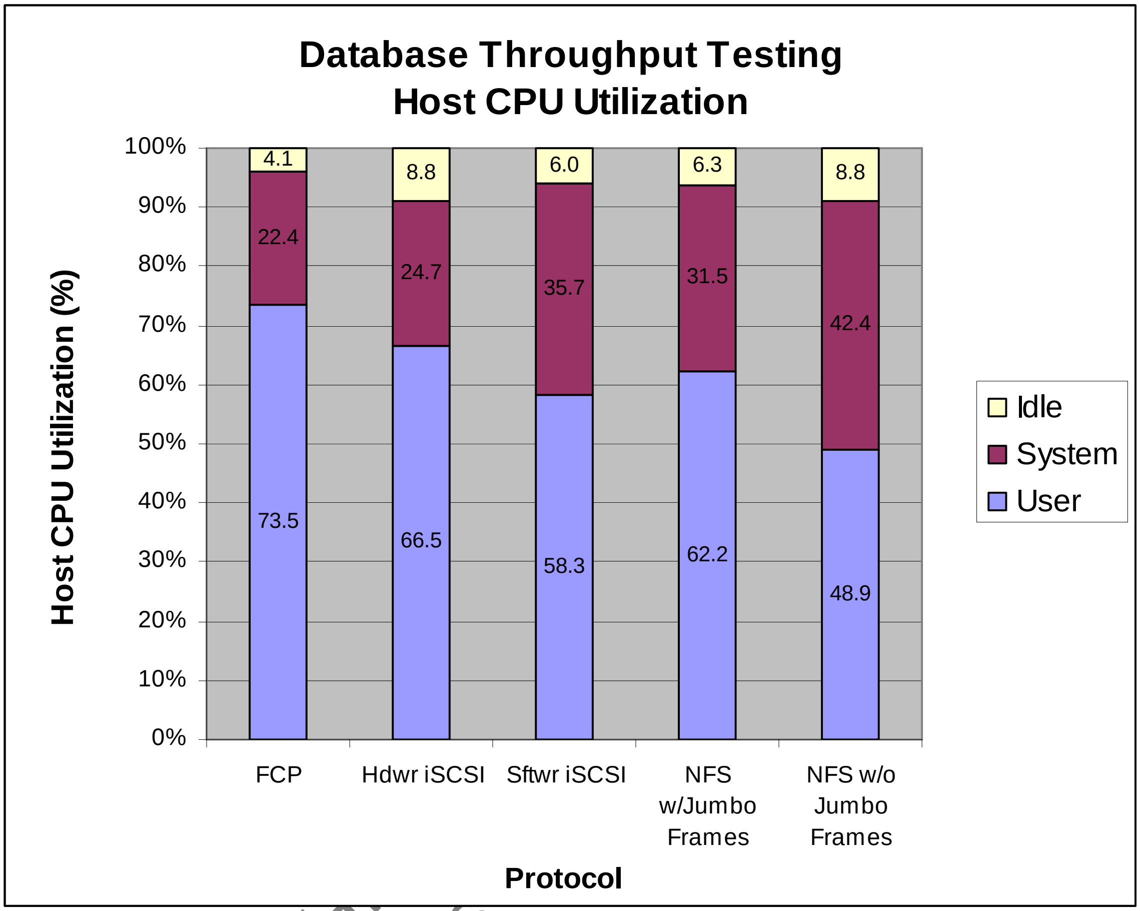

Figure 2) Host CPU utilization for database throughput tests.

Raw I/O Throughput Test Description and Results

In addition to database throughput testing, tests were also performed using sio, a NetApp I/O generator used for generating I/O loads with variable read/write, thread count, block size, and randomness characteristics. The I/O load was directed to a 20GB file on a single NetApp storage system. The 20GB file size was chosen to avoid file caching. To further ensure no caching, the following steps were performed between tests:

1. The test file system was unmounted.
2. The storage system was rebooted.
3. The test file system was remounted.

These tests were included to establish an I/O baseline for raw I/O. For this exercise, loads with several different characteristic combinations were tested, including the following:

- 80% read, 90% random, 8K block size, 96 threads
- 80% read, 90% random, 8K block size, 128 threads

The results are summarized in Figure 3 below:

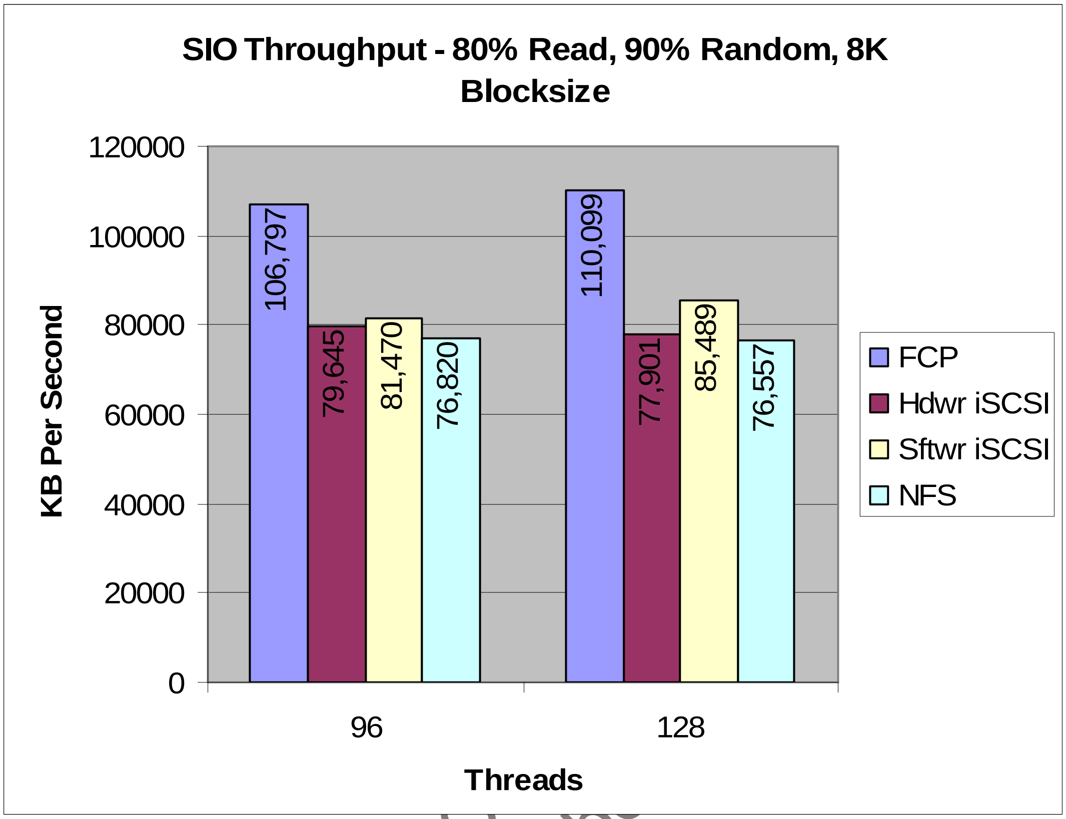

Figure 3) Raw I/O throughput for database configurations tested.

It's no surprise that FCP throughput was the highest. The fact that software iSCSI performed better than hardware iSCSI can be explained as follows:

- Without a database load, the Solaris host CPU was no longer a bottleneck, leaving more than enough CPU resources for the load generator and the software iSCSI initiator.
- The software initiator used Gigabit Ethernet NICs that were configured to use jumbo frames (MTU setting of 9,000). The QLA4010 iSCSI HBA does not have that capability. The use of jumbo frames improves throughput by allowing the movement of more data per frame, resulting in fewer interrupts.

Full Table Scan Tests with Oracle Parallel Query – Description and Results

In terms of database performance, full table scans should be avoided whenever possible. Sometimes, however, even the most well-written application running on a well-tuned database will require them. The use of high-speed storage, such as NetApp storage, can improve full table scan performance to some degree; however, there is another tool, Oracle Parallel Query, which can provide significant improvement. Oracle Parallel Query improves performance by generating a number of lightweight processes to carry out the full table scan. The number of processes generated is determined by the Degree Of Parallelism. To demonstrate this, an Oracle table with 13,400,000 rows was created in a data file located on a Network Appliance storage system.

Then a “select count(*)” command was issued against that table. The test was performed several times with the following configurations and settings:

- Parallel query not enabled with db_file_multiblock_read_count set to values of 8, 32, and 64.
- Parallel query enabled with Degree Of Parallelism (DOP) set to 32 and db_file_multiblock_read_count values of 8, 32, and 64.
- Parallel query enabled with Degree Of Parallelism (DOP) set to 64 and db_file_multiblock_read_count values of 1, 32, and 64.

Test results showed a large reduction in the total time required for the full table scan (44% reduction) simply by increasing the db_file_multiblock_read_count from 8 to 32. Tremendous improvement (as much as 94% reduction in read time) was obtained by invoking Parallel Query with a DOP of 32. Increasing the DOP from 32 to 64 did not provide any additional improvement. A summary of test results is presented in Figure 4 below:

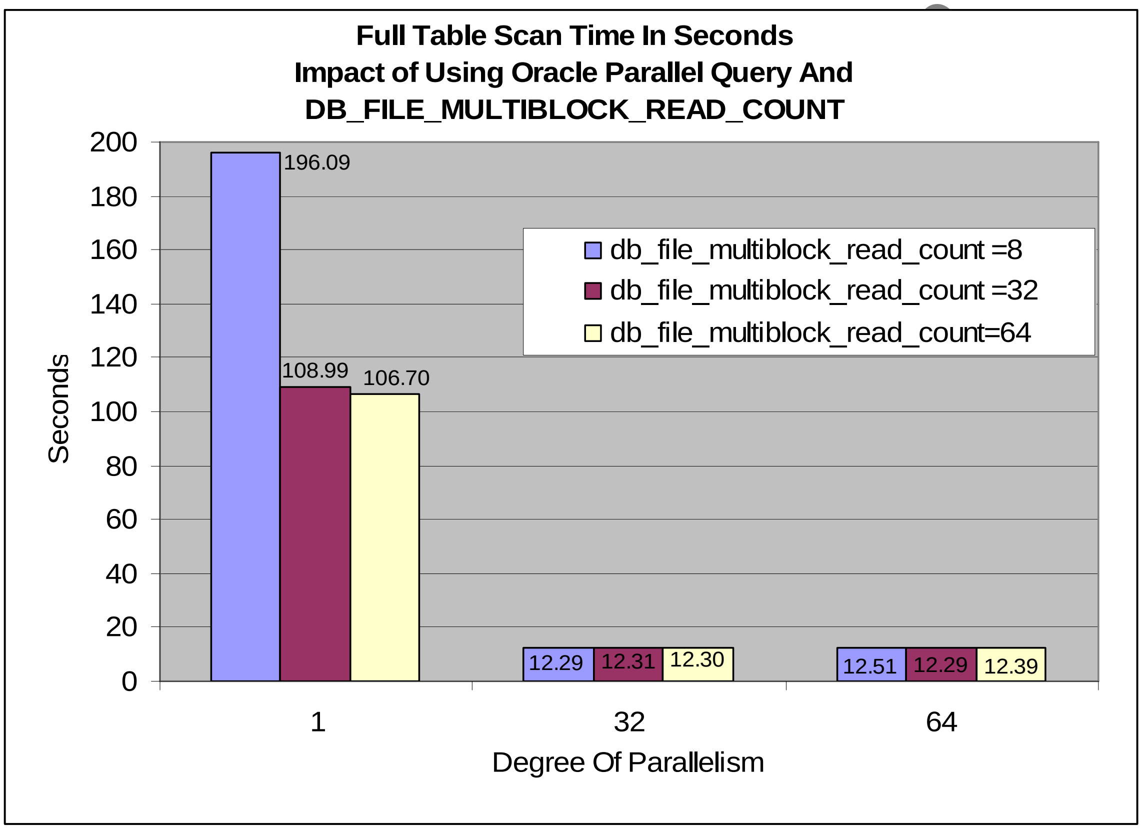

Figure 4) Full Table Scans – impact of increasing db_file_multiblock_read_count and invoking Oracle Parallel Query (Degree Of Parallelism of “1” indicates Parallel Query was not enabled).

Details of how Oracle Parallel Query was configured and applied are included in Appendix A.

Section 2: Details - FCP Environment

Database Layout

For database storage we created a single disk aggregate of 98 disks on each of two NetApp storage systems. RAID-DP was used with a RAID group size of 14 disks. The disks used were 10,000 RPM 72GB disks. In each aggregate two flexible volumes were created, one for data files and one for log files and control files. We sized the data volumes at 2,240 GB and the log volumes at 96 GB. A 20GB log LUN was created in each of the two log volumes. In each of the two data volumes we created a 200GB LUN and an 800GB LUN. A UFS file system was created on each LUN. Online redo log files and control files were stored in the 20GB file systems and Oracle data files were spread across the four data LUNs.

Storage Network Configuration

We configured the Solaris host with two QLogic QLA2342 FCP HBAs, one dedicated to each of the two storage systems. Each storage controller was configured with a corresponding QLA2342 FCP target adapter. LUN queue depth (execution-throttle) was set to 128 and the link speed autonegotiated to 2Gb/sec. Since there were only three LUNs per target, the QLogic max_luns_per_target setting was left unchanged from the default of “8.” See Figure 5 below for a network diagram.

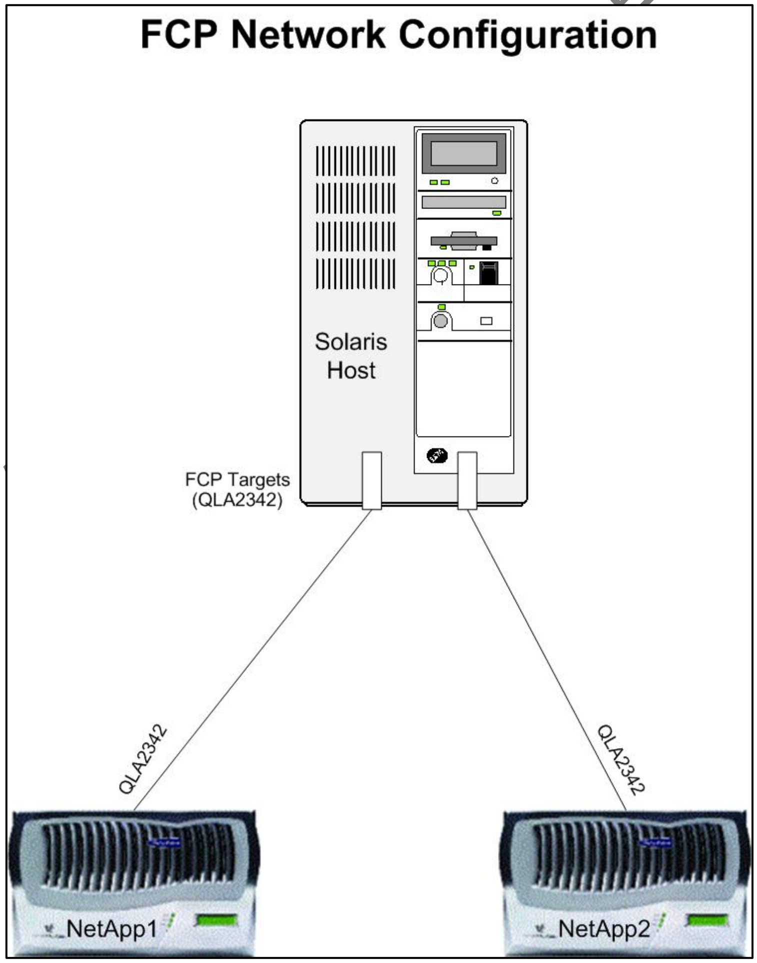

Figure 5) FCP network diagram.

Section 3: Details of Hardware iSCSI Environment

Database Layout

The hardware iSCSI volume and LUN configurations were identical to those used in the FCP test environment described in Section 2 above, as was the database layout.

Storage Network Configuration

The Solaris host was configured with two QLogic QLA4010 iSCSI HBAs, one dedicated to each of the two storage systems. Each storage controller was configured with a corresponding Emulex LP100i-D1 iSCSI target adapter. LUN queue depth (execution throttle) was set to 128 and the link speed autonegotiated to 1Gb/sec. Since there were only three LUNs per target, the QLogic max_luns_per_target setting was left unchanged from the default of “8.” See Figure 6 below for a network diagram.

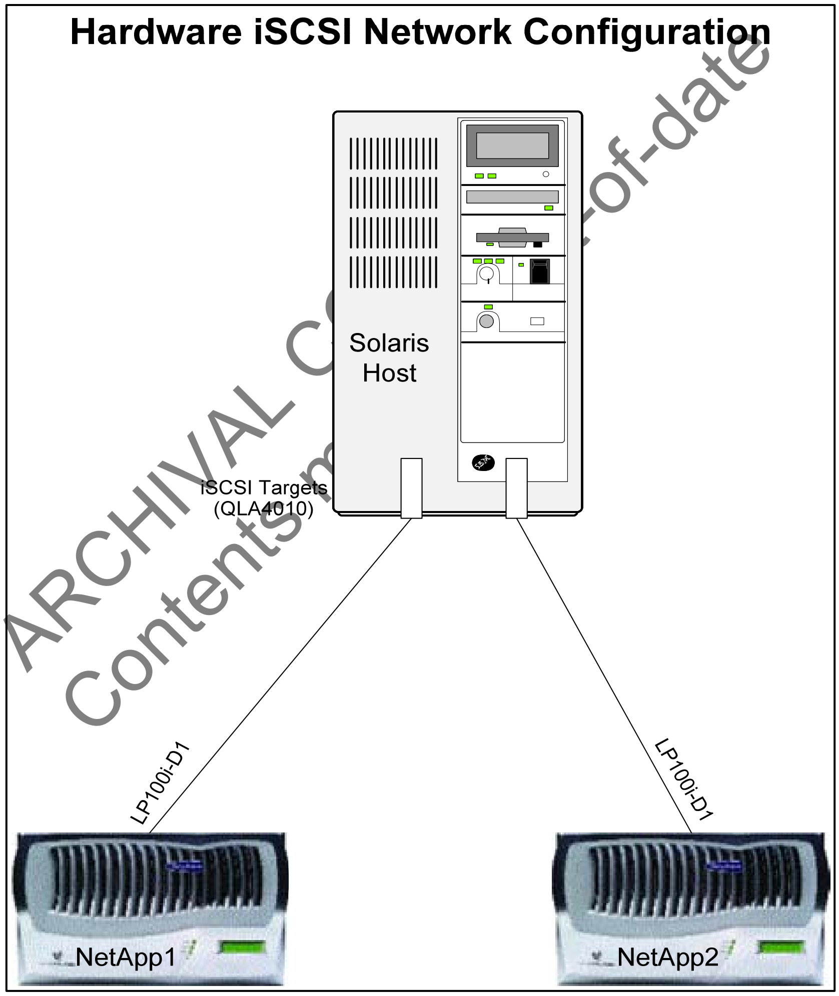

Figure 6) Hardware iSCSI network diagram.

Section 4: Details of Software iSCSI Environment

Database Layout

The software iSCSI volume and LUN configurations were identical to those used in the FCP test environment described in Section 2 above, as was the database layout.

Storage Network Configuration

The Solaris host was configured with two Intel Pro/1000 Gigabit Ethernet NICs, one dedicated to each of the two storage systems. Each storage controller was configured with a corresponding Intel Pro/1000 GbE NIC. The link speed was set to 1Gb/sec with jumbo frames enabled (MTU size of 9,000). See Figure 7 below for a network diagram.

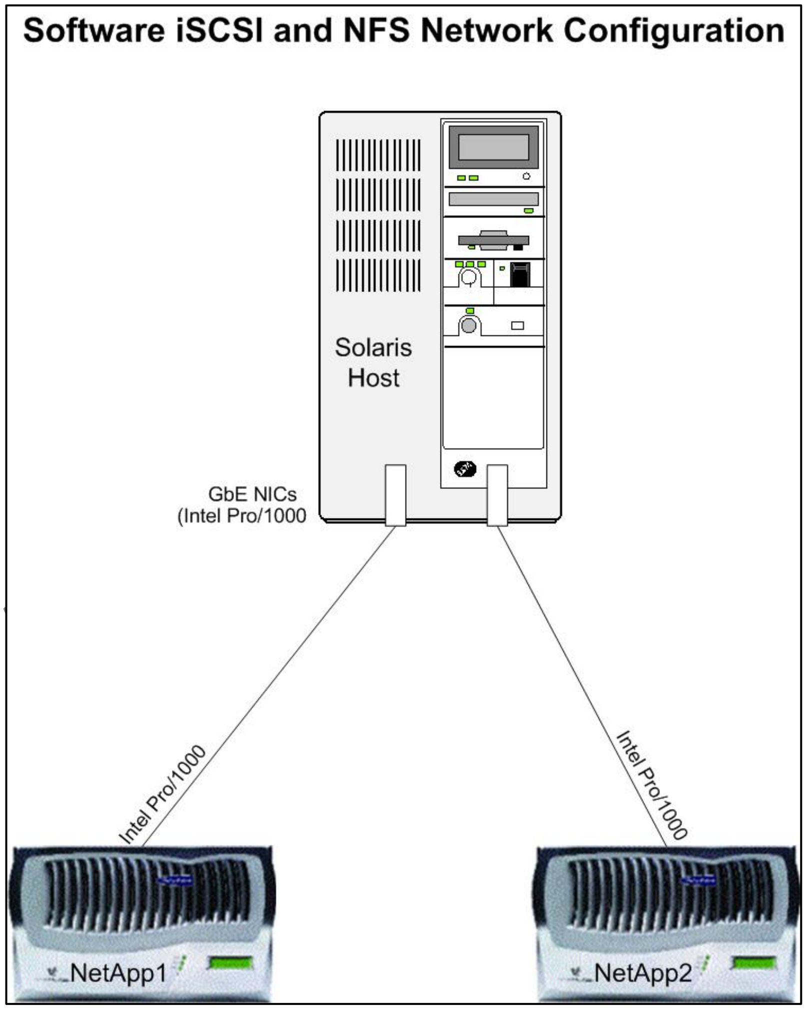

Figure 7) Software iSCSI and NFSv3 network diagram.

Section 5: Details of NFSv3 Environment

Database Layout

For database storage a single disk aggregate of 98 disks was created on each of two NetApp storage systems. RAID-DP was used with a RAID group size of 14 disks. The disks used were 10,000 RPM 72GB disks. We created two flexible volumes in each aggregate, one for data files and one for log files and control files. The data volume was sized at 2,240 GB and the log volume was sized at 96 GB. Online redo log files and control files were stored in the 96GB volumes and Oracle data files were spread across the 2,240GB volumes. The following NFS mount options were used for both data files and online redo logs:

```
rw,rsize=32768,wsize=32768,vers=3,forcedirectio,bg,hard,intr,proto=tcp
```

Storage Network Configuration

The Solaris host was configured with two Intel Pro/1000 Gigabit Ethernet NICs, one dedicated to each of the two storage systems. We configured each storage controller with a corresponding Intel Pro/1000 GbE NIC. The link speed was set to 1Gb/sec and jumbo frames were enabled with an MTU size of 9,000. See Figure 7 above for a network diagram.

Section 6: Conclusion

The goal of this project has been to provide relevant information for making informed, intelligent decisions in the architecture of enterprise data systems utilizing Oracle 10*g* with the Solaris 10 operating system. That has been accomplished through the tests described in this document. We do recommend that the data outlined herein be used for comparison purposes and the individual data points not be considered as absolutes. It must be recognized that enterprise data systems can vary greatly in terms of complexity, configuration, and application workload, impacting both performance and functionality.

Additional details of test procedures and test environments are included in the appendix portion of this document.

Appendix A: Oracle Parallel Query – Syntax and Application

Below is an example of the commands used for enabling and using Parallel Query in performing a full table scan on a test table:

```
alter table test_table parallel(degree 32);
select /*+ PARALLEL(test_table 32) */ count(*) from test_table;
```

where “test_table” is the table name and “32” is the degree of parallelism. Please note the select statement uses a hint to force Parallel Query for the count(*) command.

In the same fashion, the following commands were used for executing the table scan without Parallel Query:

```
alter table test_table noparallel;
select count(*) from test_table;
```

Appendix B: Pertinent Solaris Kernel Parameter Settings with Syntax

```
set nfs:nfs3_max_threads=64
set rlim_fd_cur=1024
set sq_max_size=100
ndd -set /dev/tcp tcp_recv_hiwat=65535
ndd -set /dev/tcp tcp_xmit_hiwat=65535
```

Appendix C: Pertinent NetApp Storage System Settings

```
nvfail          on
fcp.enable      on
iscsi.enable    on
nfs.v3.enable   on
nfs.tcp.enable  on
nfs.tcp.recvwindowsize  65536
nfs.tcp.xfersize        65536
iscsi.iswt.max_ios_per_session  256
iscsi.iswt.tcp_window_size      262800
iscsi.max_connections_per_session      16
```

Appendix D: Mount Options

UFS Mount Options (/etc/vfstab entries)

```
forcedirectio, noatime
```

NFS Mount Options (/etc/vfstab entries)

```
rw,rsize=32768,wsize=32768,vers=3,forcedirectio,bg,hard,intr,proto=tcp
```

Appendix E: Patches, Drivers, and Software

Fujitsu Server

Solaris 10 with November 2005 patch cluster

NetApp Storage System OS

Data ONTAP 7.1RC4

QLA2342 FCP Adapters

Driver Version v.5.00

QLA4010 iSCSI Adapters

Driver Version v.4.01

Intel PRO/1000 GbE Network Interface

Driver Version 11.10.0,REV=2005.09.15.00.13

Sun iSCSI Software Initiator

SUNWiscsir Sun iSCSI Device Driver
Version: 11.10.0,REV=2005.01.04.14.31
SUNWiscsiu Sun iSCSI Management Utilities
Version: 11.10.0,REV=2005.01.04.14.31

Appendix F: Pertinent Oracle Parameter (pfile) Settings

OLTP Throughput Tests

compatible	10.2.0.1.0
cursor_space_for_time	TRUE
db_16k_cache_size	3154116608
db_block_size	8192
db_cache_size	23068672000
db_files	600
filesystemio_options	setall
db_writer_processes	4
java_pool_size	16777216
job_queue_processes	16
log_buffer	268419072
processes	900
shared_pool_size	536870912
statistics_level	typical
transactions	900
undo_management	AUTO
undo_retention	0
_log_parallelism	4
_kgl_large_heap_warning_threshold	4194304

Parallel Query Tests

compatible	10.2.0.1.0
parallel_max_servers	100
parallel_min_servers	64
db_files	600
db_file_multiblock_read_count	[8,32,64]
db_cache_size	300M
java_pool_size	16777216
db_16k_cache_size	200M
filesystemio_options	setall
statistics_level	basic

log_buffer	1048576
_log_parallelism	4
processes	900
transactions	900
shared_pool_size	150M
cursor_space_for_time	TRUE
db_writer_processes	4
db_block_size	8192
job_queue_processes	16
undo_management	auto
undo_retention	0
_kgl_large_heap_warning_threshold	4194304

Appendix G: Lessons Learned, Bugs, etc.

Oracle Parameter "_kgl_large_heap_warning_threshold"

The following warning message appeared repeatedly in the alert log during initial testing: "Heap size 2294K exceeds notification threshold (2048K)." We found the problem documented in Oracle Metalink Note: 330239.1. The problem occurs only in Oracle 10.2.0.1.0 and is the result of the KGL heap warning threshold default setting being too low. The solution was to set the _kgl_large_heap_warning_threshold parameter to 4194304.

Solaris 10 UFS LUN Misalignment Problem

Beginning with Solaris 9 Update 4, multi-terabyte UFS has shipped with Solaris by default. As a result EFI labels are required instead of SMI labels. EFI labels occupy the first 34 sectors of a disk or LUN, resulting in the starting sector of the LUN not beginning on a 4K boundary. With WAFL®, LUNs need to be aligned with 4K boundaries to avoid partial writes and reads. Partial reads and writes cause the storage system to perform considerably more work, degrading system performance. To avoid this problem, partition 0 must be sized to 39 cylinders to cause the remaining LUN(s) to start on a 4K boundary. Example procedure:

```
format -e
Specify disk (enter its number):          enter the number corresponding to the
disk you wish to partition
[disk formatted]
Disk not labeled.  Label it now?  no
format> partition
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1
Do you wish to continue creating a new partition
table based on above table[yes]?  yes
Free Hog partition[6]? 6
Enter size of partition 0 [0b, 33e, 0mb, 0gb, 0tb]: 39e
Enter size of partition 1 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Enter size of partition 2 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Enter size of partition 3 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Enter size of partition 4 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Enter size of partition 5 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Enter size of partition 7 [0b, 33e, 0mb, 0gb, 0tb]: 0b
Ready to label disk, continue? yes
partition> quit
format> quit
```

Resolving this problem by using the above workaround resulted in a database throughput increase in the range of 12–14%. We are actively pursuing resolution of this problem with Sun.

Appendix H: Pertinent Hardware Details

Server: Fujitsu PrimePower 650 8-way server running Solaris 10 and Oracle 10.2.0.1.0
(8) 810 MHz CPUs
32GB RAM
Storage System: Two NetApp storage systems running Data ONTAP 7.1RC4
FAS980 with (98) 72GB 10K RPM disks each

Appendix I: Adherence To Best Practices

Every effort was made to conform to Oracle/NetApp best practices as they were defined at the time the tests described in this paper were performed. It must be understood that occasionally best practices do change; therefore, any discrepancy between this document and published best practices should be resolved in favor of the published best practices documentation.

Acknowledgements

Technical Direction and Advice

Glenn Colaco
Performance Engineer – Performance Analysis Team
Network Appliance, Inc.

Stephen Daniel
Director Database Platforms and Performance Technology
Network Appliance, Inc.

References

“Database Performance with NAS: Optimizing Oracle with NFS”
http://www.netapp.com/library/tr/3322.pdf

“System Administration Guide: Devices and File Systems”
http://docs.sun.com/app/docs/doc/819-2723/

“Oracle 10g for UNIX: Integrating with a Network Appliance Filer”
http://www.netapp.com/library/tr/3353.pdf

Network Appliance Technical Library
http://www.netapp.com/library/tr

NetApp

www.netapp.com