



Technical Report

AIX Protocol Performance Comparison with Oracle Database 11g Release 2

Freddy Grahn, NetApp
October 2010 | TR-3871

A PERFORMANCE COMPARISON

This technical report compares the performance of the different network protocols and filesystems and how they interact with an Oracle® Database 11g Release 2 OLTP, running on AIX 6.1, using 10GbE and 4GB Fibre Channel (FC).

TABLE OF CONTENTS

1	INTRODUCTION AND EXECUTIVE SUMMARY	4
1.1	INTRODUCTION.....	4
1.2	EXECUTIVE SUMMARY.....	4
2	RESULTS AND ANALYSIS.....	5
2.1	OLTP WORKLOAD	5
2.2	SUPPORTING ANALYSIS.....	6
3	CONFIGURATION DETAILS.....	9
3.1	GENERIC CONFIGURATION	9
3.2	KNFS AND ORACLE DNFS	9
3.3	FIBRE CHANNEL PROTOCOL WITH IBM JFS2 AND ORACLE ASM	13
4	CONCLUSION.....	15
5	APPENDIXES.....	15
5.1	HARDWARE.....	15
5.2	STORAGE LAYOUTS.....	16
5.3	ORACLE INITIALIZATION PARAMETERS	19
5.4	AIX KERNEL PARAMETERS.....	21
5.5	OTHER AIX OS SETTINGS.....	22
6	ACKNOWLEDGMENTS.....	22

LIST OF TABLES

Table 1)	Hardware used for kNFS and DNFS.	10
Table 2)	Hardware used for JFS2 and ASM.....	13
Table 3)	AIX host.	15
Table 4)	NetApp controllers.	16
Table 5)	Oracle initialization parameters (init.ora).....	19
Table 6)	AIX nondefault kernel parameters.	21
Table 7)	AIX shell limits for Oracle.	22

LIST OF FIGURES

Figure 1)	Application throughput comparison.	5
Figure 2)	OETs and DB file sequential read latencies.....	7
Figure 3)	OETs and CPU by protocol.	8
Figure 4)	CPU breakdown by protocol.....	9
Figure 5)	kNFS and DNFS network.	10
Figure 6)	JFS2 and ASM network.....	13

Figure 7) kNFS and DNFS volume layout.	17
Figure 8) JFS2 volume layout.	18
Figure 9) ASM volume layout.	19

1 INTRODUCTION AND EXECUTIVE SUMMARY

1.1 INTRODUCTION

To help NetApp's customers select the best solutions for their IT infrastructure, NetApp provides up-to-date collateral on its products. This technical report contains updated performance information and tuning recommendations for Oracle Database 11g Release 2 on AIX 6.1 with NetApp® storage systems. This information will help customers to make informed decisions about the optimal storage protocol for their Oracle database deployments.

The protocols used in this technical report are kernel Network File System (kNFS) and Oracle's new Direct NFS (DNFS) running over 10GbE connections, as well as IBM Enhanced Journaled File System (JFS2) and Oracle Automatic Storage Management (ASM) running over 4GB Fibre Channel (FC).

This report makes a distinction between two different network NFS clients. The NFS client provided by the AIX operating system is referred to as kernel NFS, or kNFS. The other NFS client, referred to as Oracle Direct NFS or DNFS, is a new NFS client provided by Oracle 11g R2; it runs as part of the Oracle database. DNFS can be used only to access the Oracle database files.

The file systems used in this technical report are NFS, IBM's JFS2, and Oracle's ASM.

Anyone who is in the process of selecting a storage protocol or file system for their Oracle database running on NetApp storage, or anyone who is curious about how the different protocols compare in performance, should read this report. This report can also be used as a performance-tuning reference.

1.2 EXECUTIVE SUMMARY

We tested supported protocols for the Oracle database to demonstrate the level of performance that each protocol can offer. These protocols included kNFS, DNFS, and Fibre Channel Protocol (FCP). Data ONTAP® 8.0 7-Mode, the latest release in the Data ONTAP family, was used in conjunction with 64-bit aggregates to demonstrate the performance capabilities it offers our customers. For each of the NFS categories (kNFS and DNFS), we tested with and without jumbo frames to examine the performance improvements and CPU savings when using jumbo frames. The FC tests used a single 4GB IBM-branded host bus adapter (HBA), and the Ethernet tests used a single 10GbE adapter.

An OLTP workload was used for all performance tests. This workload simulated 150 users interacting with 8,000 product warehouses in an order-processing application. The client processes for the OLTP application were executed on a separate application server (client-server mode). The number of order entry transactions (OETs) completed per minute was the primary metric used to measure application throughput. The I/O mix for the OLTP workload was approximately 65% reads and 35% writes.

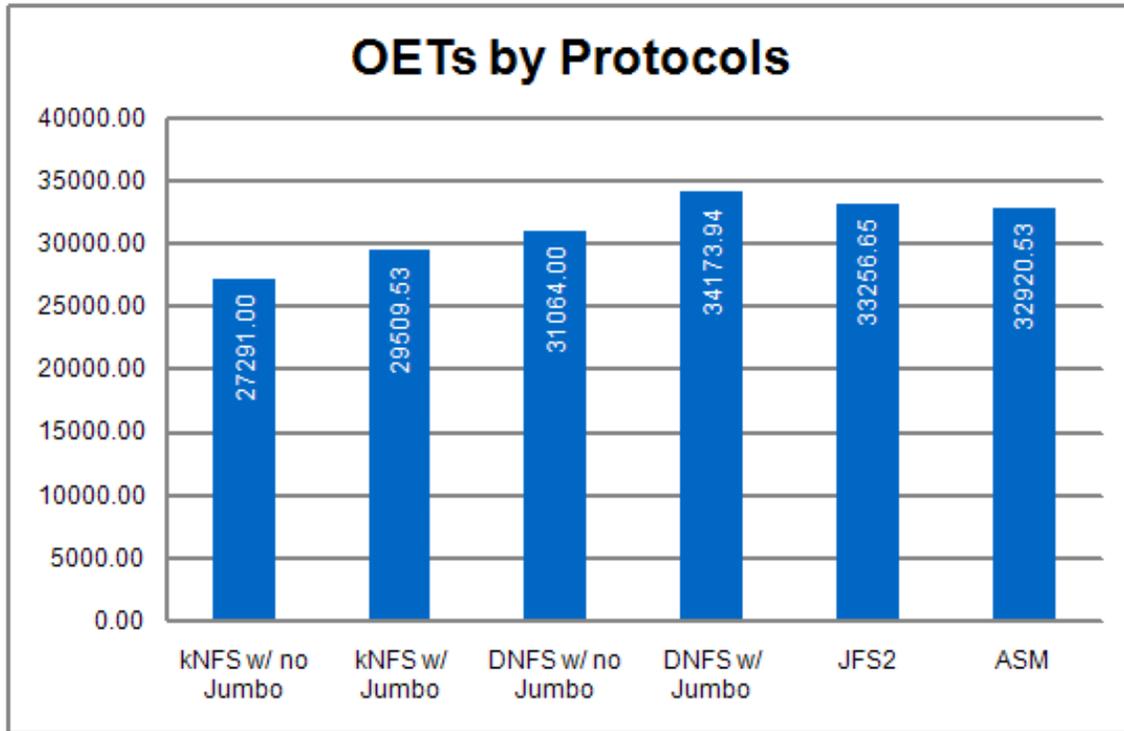
The OLTP workload used for these tests was input/output operations per second (IOPS) using an 8k request size; it did not generate sufficient throughput to saturate any of the network connections. As a result, we do not expect the tests that used 10GbE connections to deliver better performance compared to those using 4GB FC connections simply due to the differences in the connections' total bandwidth.

To be sure that our test results were consistent and that our comparison was valid, we used the same workload for all comparisons. To determine the workload, we used a base configuration of an Oracle database using kNFS without jumbo frames with a targeted latency of 10ms. Once determined, this load was held constant across the other protocols in order to have an apples-to-apples comparison of application throughput. The bottleneck in these tests never existed at the storage layer.

Figure 1 shows that application throughputs for all protocols were excellent. Looking first at application throughput for kNFS without jumbo frames, we observed throughput of 27291.00 OETs per minute. Enabling jumbo frames with kNFS resulted in an 8% increase in overall performance, to 29509.53 OETs per minute. Throughput for Oracle's DNFS without jumbo frames was 31064 OETs per minute, which improved approximately 10%, to 34174 OETs per minute, after enabling jumbo frames. Performance

using Oracle's ASM and IBM's JFS2 over FCP was 32920.53 and 33256.65 OETs per minute, respectively.

Figure 1) Application throughput comparison.



The Oracle DNFS client that runs as part of the Oracle11g R2 delivered consistently higher performance than kNFS with or without jumbo frames. It is an excellent choice for customers who want to obtain FCP-like performance for OLTP workloads at NFS costs. We found that using DNFS with jumbo frames generated approximately 3% better performance as compared to either of the options tested—using either ASM or JFS2 with FCP.

2 RESULTS AND ANALYSIS

Before analyzing the test results, it is important to understand the testing methodology and the workload employed. A consistent testing methodology was used across all storage protocols. This methodology used an OLTP workload to demonstrate the capabilities of each protocol. The same database was used for both the kNFS and DNFS protocols, and two other duplicate databases were used for the ASM and JFS2 tests.

2.1 OLTP WORKLOAD

The database created for the OLTP workload uses a data model designed for order entry transaction processing. The OLTP database was approximately 1.03TB in size and contained approximately 8,000 warehouses. For testing across all protocols, we used a workload that simulated 150 users and 8,000 active warehouses. The client processes for the OLTP application were executed on a separate application server (client-server mode). The load was configured to saturate host-side resources for the kNFS protocol. In these tests, the CPU was the host-side resource that became a bottleneck for kNFS. The OLTP load (150 users and 8,000 active warehouses) was held constant across the other protocols in

order to have an apples-to-apples comparison for application throughput. There was no bottleneck in these tests on the NetApp storage platform.

A mix of different types of transactions was used during each OLTP test run. These transaction types included order entries, payments, order status, delivery, and stock level.

The number of OETs completed per minute was the primary metric used to measure application throughput.

The I/O mix for the OLTP workload was approximately 65% reads and 35% writes.

2.2 SUPPORTING ANALYSIS

Figure 2 shows the total number of OETs and the DB file sequential read wait event as reported by the Oracle database for each test. The DB file sequential read wait event indicates the amount of time, in milliseconds, that it took to read a single database block from storage. We observed the following results:

- kNFS configurations generated the lowest performance and highest latencies compared to the other protocol configurations tested.
- Enabling jumbo frames with kNFS resulted in an increase of approximately 8% in OETs, with an associated reduction in latency of about 10% compared to using kNFS without jumbo frames.
- Employing DNFS delivered improved performance and similar to lower latencies compared to kNFS.
- Enabling jumbo frames with DNFS generated the best overall OET values of all protocol configurations tested.
- FCP configurations using either JFS2 or ASM delivered comparable performance, while JFS2 resulted in noticeably lower latencies compared to all other protocol configurations tested.

Figure 2) OETs and DB file sequential read latencies.

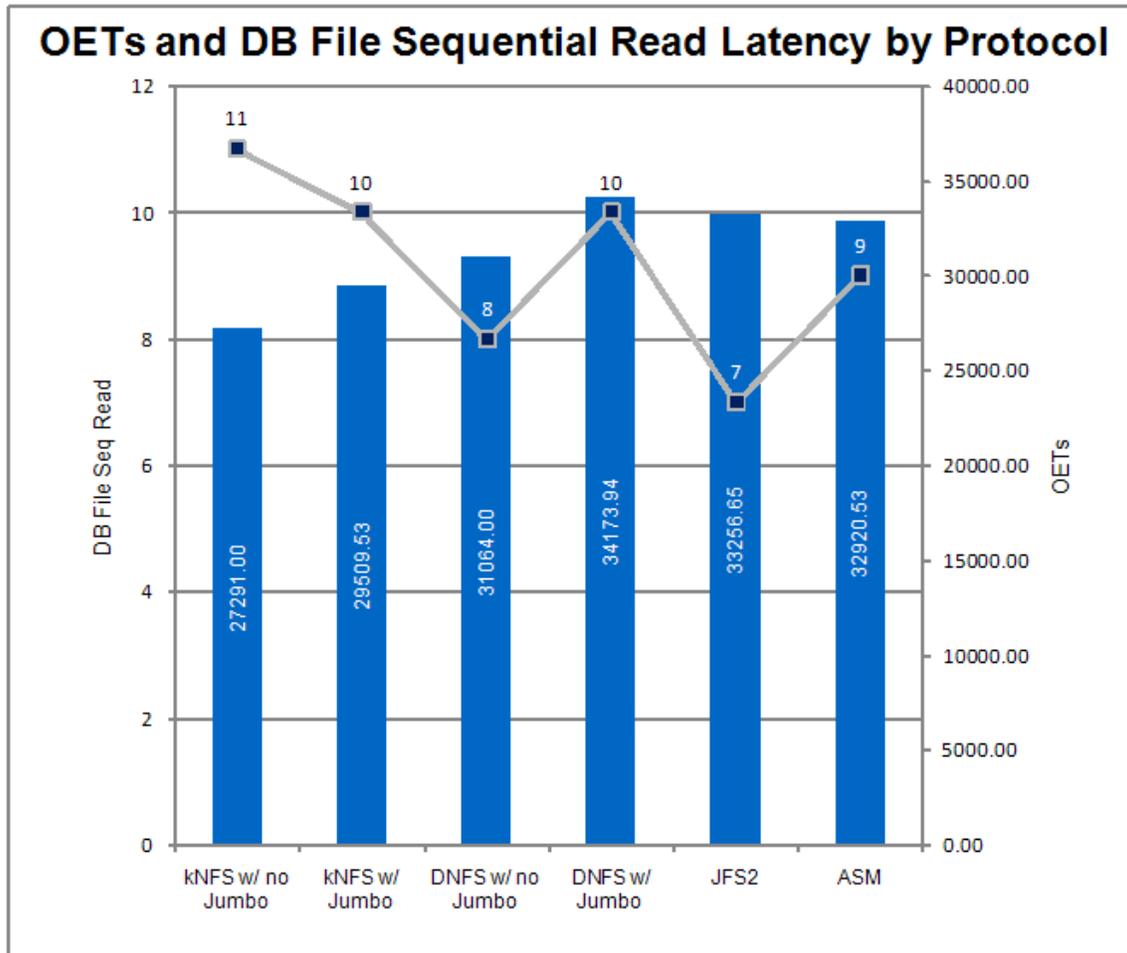


Figure 3 shows the typical amount of host CPU used to generate the corresponding OETs per minute for each protocol. We observed the following results:

- Using kNFS without jumbo frames consumed more host CPU resources than any of the other test configurations.
- Enabling jumbo frames with kNFS reduced the host CPU overhead slightly compared to kNFS without jumbo frames, while allowing the overall OETs to increase by approximately 8%.
- Using DNFS without jumbo frames resulted in better performance compared to either kNFS configuration, while consuming similar levels of host CPU resources.
- Enabling jumbo frames with DNFS reduced the host CPU use by approximately 10% compared to DNFS without jumbo frames configured, while allowing approximately 10% improvement in OETs.
- Using either of the FCP options (JFS2 or ASM) required noticeably lower amounts of host CPU resources, while delivering comparable to slightly lower performance compared to DNFS with jumbo frames enabled. This is because all storage communications and protocol processing are offloaded onto the FC HBA, freeing up the host CPU.
- All software based protocols placed an additional burden on the host CPU compared to FCP.

Figure 3) OETs and CPU by protocol.

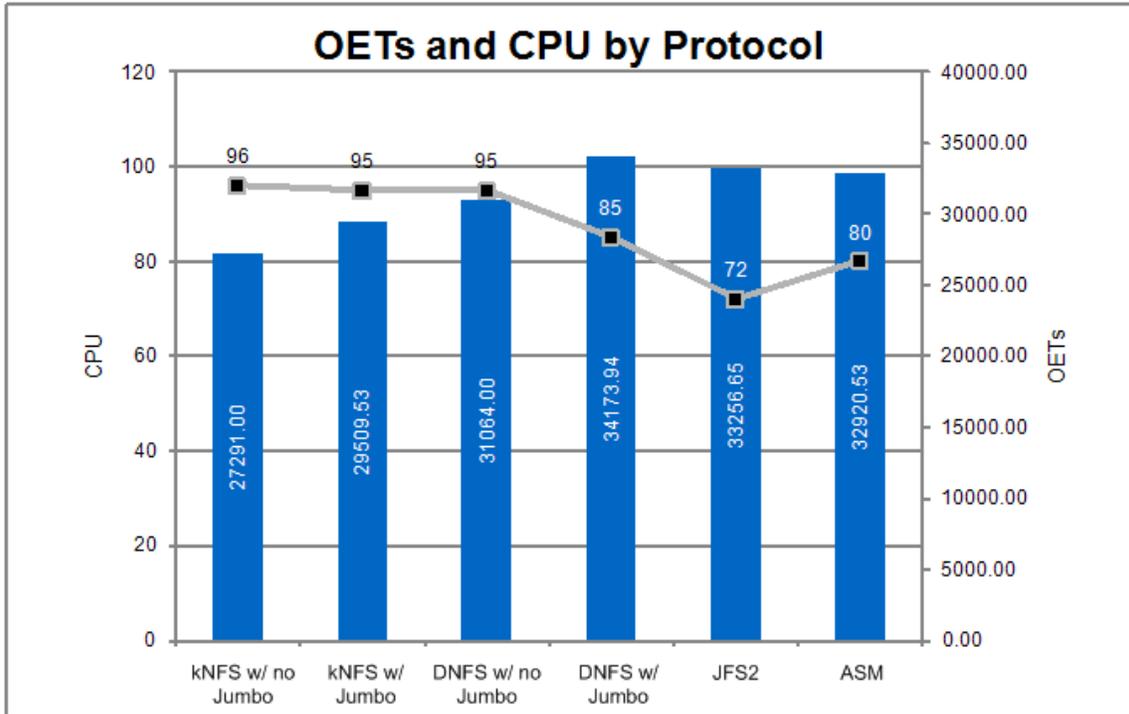
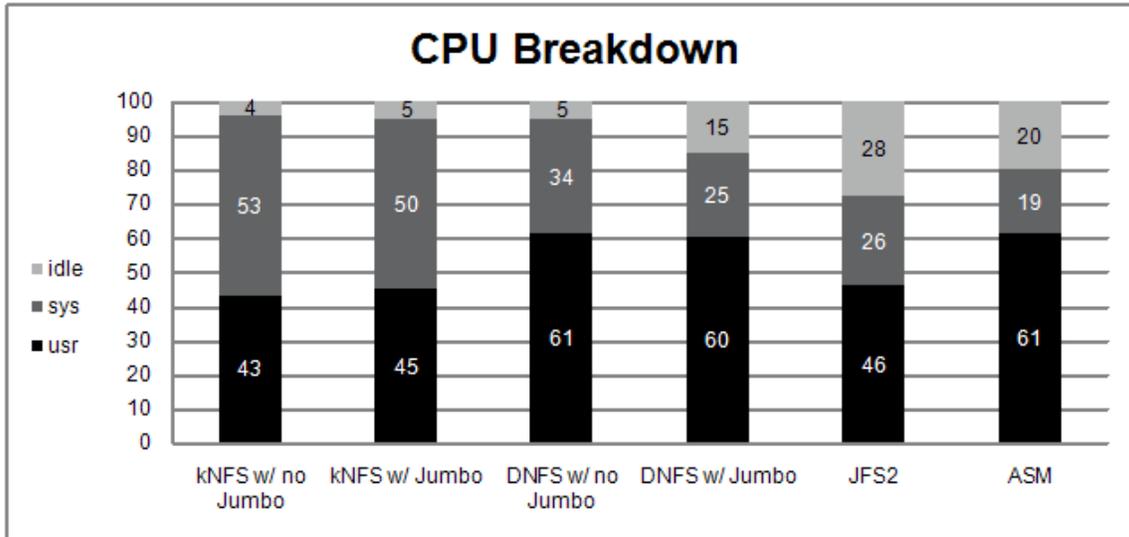


Figure 4 shows the host CPU resources consumed by each protocol combination. Regardless of whether or not jumbo frames are enabled, kNFS consumes the largest portion of system CPU because the NFS client runs in the AIX kernel. Although Oracle DNFS is another NFS client, it consumes a much smaller portion of the system CPU compared to kNFS. This is because the DNFS client runs within the user space of the OS and therefore the user portion of the CPU usage is larger than kNFS. Another reason the user portion of the CPU is greater for DNFS than for kNFS is that DNFS completed more transactions per minute, resulting in a higher number of OETs. Also note that using DNFS with jumbo frames reduced the amount of system CPU resources by approximately 26% compared to DNFS without jumbo frames enabled, demonstrating that enabling jumbo frames with DNFS can result in improved performance and more efficient use of host CPU resources.

As is typical of the two FCP tests, JFS2 and ASM used the smallest amount of system CPU because the storage communication and protocol processing were offloaded onto the HBA. Compared to ASM, JFS2 consumed approximately 25% less user-level CPU and approximately 36% more system-level CPU.

Figure 4) CPU breakdown by protocol.



3 CONFIGURATION DETAILS

3.1 GENERIC CONFIGURATION

Where possible, the test environments used configurations that were as similar as possible, including:

- Identical hardware
- Similar storage layouts
- Similar Oracle initialization parameters
- Similar AIX OS settings

In the following sections, when a configuration deviates from the norm or is exclusive to a particular protocol, that configuration setting is explained in detail.

3.2 KNFS AND ORACLE DNFS

NETWORK DIAGRAM

Figure 5 shows a kernel NFS and a direct NFS using one 10GbE link. Table 1 describes the storage network hardware depicted in the diagram.

Figure 5) kNFS and DNFS network.

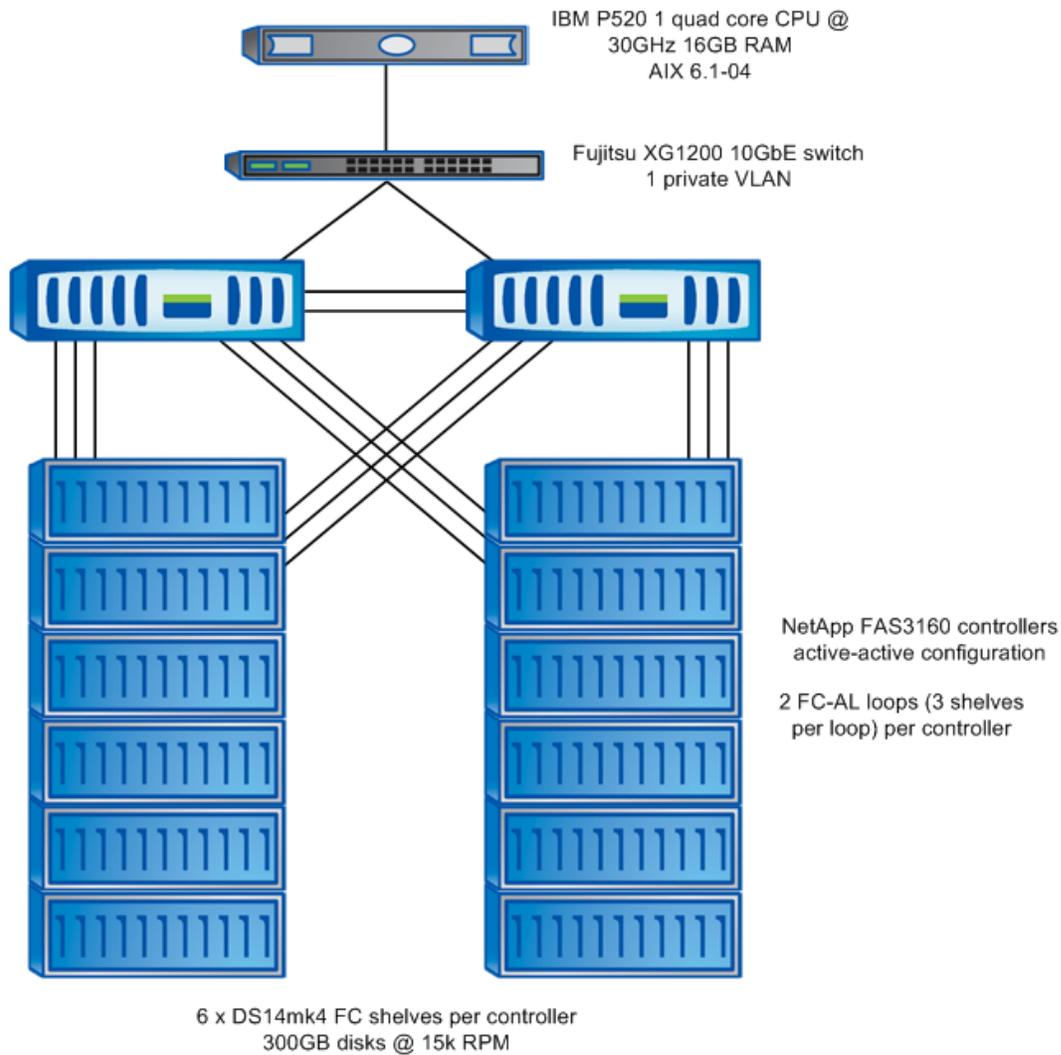


Table 1) Hardware used for kNFS and DNFS.

Hardware	Description
Database Server	IBM P520 1 Quad Core CPU @ 3.0GHz, 16GB RAM, AIX 6.1-04, 1 single-port 10GbE card
Switch	Fujitsu XG1200 10GbE, 1 private VLAN
NetApp Controllers	FAS3160 controllers, active-active configuration; 2 FC-AL loops (3 shelves per loop) per controller, 1 single-port Chelsio 10GbE card (each storage controller)
Storage Shelves	6 x DS14mk4 FC shelves per controller, 300GB disks @ 15k RPM

STORAGE NETWORK CONFIGURATION

Jumbo frames were used in this network configuration for the two jumbo frame tests. During these tests, a maximum transmission unit (MTU) size of 9,000 was set for all storage interfaces on the host, for all interfaces on the NetApp controllers, and for the ports involved on the switch. For the nonjumbo frame tests, we configured the MTU size to 1,500 for all of the previously noted ports and interfaces. We used a single private subnet to segment Ethernet traffic between the host and storage controllers.

AIX NETWORK CONFIGURATION

```
nfs.tcp.xfersize 65536
```

To take advantage of our larger read size (rsize) and window size (wsize), and to enable TCP to use larger transfer sizes, we increased the AIX setting.

KNFS MOUNTS AND OPTIONS

The following excerpt is from the `/etc/filesystems` file on the AIX host.

```
/mnt/oradata:
  dev           = "/vol/oradata_nfs"
  vfs           = nfs
  nodename     = 192.168.20.20
  mount        = true
  options      = rw,bg,hard,proto=tcp,vers=3,rsize=65536,wsiz=65536,cio,timeo=600,intr
  account      = false

/mnt/oralog:
  dev           = "/vol/oralog_nfs"
  vfs           = nfs
  nodename     = 192.168.20.20
  mount        = true
  options      = rw,bg,hard,proto=tcp,vers=3,rsize=65536,wsiz=65536,cio,timeo=600,intr
  account      = false

/mnt/oradata2:
  dev           = "/vol/oradata2_nfs"
  vfs           = nfs
  nodename     = 192.168.20.40
  mount        = true
  options      = rw,bg,hard,proto=tcp,vers=3,rsize=65536,wsiz=65536,cio,timeo=600,intr
  account      = false

/mnt/oralog2:
  dev           = "/vol/oralog2_nfs"
  vfs           = nfs
  nodename     = 192.168.20.40
  mount        = true
  options      = rw,bg,hard,proto=tcp,vers=3,rsize=65536,wsiz=65536,cio,timeo=600,intr
  account      = false
```

The mount options we used came from the NetApp KB article [kb7518: "What are the mount options for databases on NetApp NFS?"](#)

We made one change to the mount options. KB article kb7518 uses an rsize and wsize of 32,768; however, we decided to use an rsize and wsize of 65,536. Using these settings enabled us to take better advantage of larger TCP window sizes, which are now available on the storage controllers.

DNFS MOUNTS AND CONFIGURATION

With Oracle DNFS, the NFS mount points and network paths are specified in a new configuration file, `oranfstab`. However, the NFS mounts must still be specified in the `/etc/fstab` file because Oracle cross-checks the entries in this file with the `oranfstab` file. If any NFS mounts do not match between `/etc/filesystems` and `oranfstab`, then DNFS does not use those NFS mount points. We

maintained the same NFS mounts as shown in the preceding excerpt from the `/etc/filesystems` file on the AIX host. The following excerpt is from the `oranfstab` file that was used during the DNFS tests.

```
server: node-c1
path: 192.168.20.20
export: /vol/oradata_nfs mount: /mnt/oradata
export: /vol/oralog mount: /mnt/oralog

server: node-c2
path: 192.168.20.40
export: /vol/oradata2_nfs mount: /mnt/oradata2
export: /vol/oralog2_nfs mount: /mnt/oralog2
```

For more information about DNS configuration, see the [Oracle Database Installation Guide](#) for Oracle11g R2.

DATA ONTAP TUNING OPTIONS

`nfs.tcp.recvwindowsize 262144`

This option increases the size of the TCP receive window for NFS. The TCP receive window on the client should be adjusted to match the size of the NetApp controller because TCP reduces the size of the larger TCP receive window to match the size of the smaller TCP window. The following section, “AIX Tuning Options,” contains additional information.

We found that it was necessary to increase the default value of 26,280 to realize the performance increase of jumbo frames.

`nfs.tcp.xfersize 65536`

This is the default setting for this option. Make sure that it is not reduced. This option controls the TCP transmit window size for NFS.

`no_atime_update on`

By default, this volume setting is set to off. There is an extremely slight overhead associated with this setting. Because Oracle does not require knowledge of the latest access time update, we disabled this volume setting on each of the data and log volumes.

`setflag waf_max_write_alloc_blocks 256`

We set one nonstandard flag in the `/etc/rc` file of each storage system. This flag optimizes the WAFL® (Write Anywhere File Layout) on-disk data layout.

AIX TUNING OPTIONS

`tcp_sendspace 262144`

`tcp_recvspace 262144`

Because the storage controller receive window size was set to 262,144, we set both the send space and receive space values to match. TCP reduces or increases the size of the window to match what is being sent from the host.

STORAGE CONFIGURATION

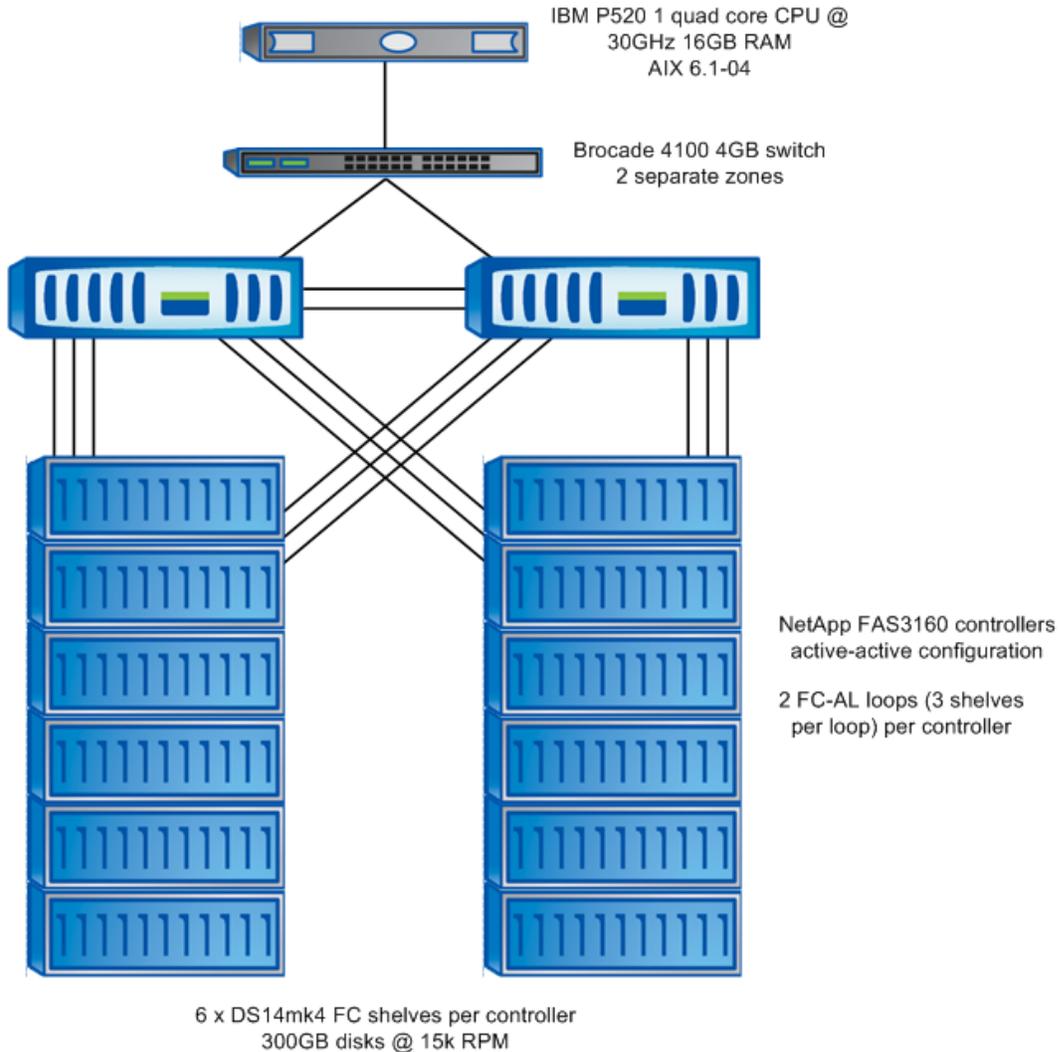
Data ONTAP 8.0 7-Mode was used for both storage controllers and all tests. The storage layout for NFS and DNFS testing is detailed in the NFS and DNFS volume layout diagram in appendix 5.2, “Storage Layouts.”

3.3 FIBRE CHANNEL PROTOCOL WITH IBM JFS2 AND ORACLE ASM

NETWORK DIAGRAM

Figure 6 shows JFS2 and ASM using one 4GB link. Table 2 describes the storage network hardware depicted in the diagram.

Figure 6) JFS2 and ASM network.



We used a combination of Oracle ASM and IBM JFS2 to test with Fibre Channel Protocol (FCP). We used both as the file system and volume manager for the logical unit numbers (LUNs) presented from the storage system. A single DATA disk group was used to evenly spread I/Os across the eight LUNs that were presented from both NetApp controllers.

Table 2) Hardware used for JFS2 and ASM.

Hardware	Description
Database Server	IBM P520 1 Quad Core CPU @ 3.0GHz, 16GB RAM, AIX 6.1-04, 1 single-port Emulex 4GB HBA (PCI-X)

Hardware	Description
Switch	Brocade 4100 4GB switch, 2 separate zones
NetApp Controllers	FAS 3160 controllers, active-active configuration; 2 FC-AL loops (3 shelves per loop) per controller, 1 built-in 4GB HBA
Storage Shelves	6 x DS14mk4 FC shelves per controller; 300GB disks @15k RPM

STORAGE NETWORK CONFIGURATION

The single target HBA from both NetApp controllers was connected to the Brocade 4100 switch. The Data ONTAP CFMODE was set to SINGLE_IMAGE. This is the default mode and is the only mode that supports 4GB speeds.

DATA ONTAP TUNING OPTIONS

`no_atime_update on`

By default, this volume setting is set to off. There is an extremely slight overhead associated with this setting. Because Oracle does not require knowledge of the latest access time update, we disabled this volume setting on each of the data and log volumes.

`setflag waf_max_write_alloc_blocks 256`

We set one nonstandard flag in the `/etc/rc` file of each storage system. This flag optimizes the WAFL on-disk data layout.

ALUA

The initiator groups (igroups) defined for the LUNs on the storage controllers were set to use Asymmetric Logical Unit Access (ALUA). ALUA is an extension of the FC protocol that defines how multipath I/O should be managed between hosts and storage devices. The standard enables the host to understand the performance difference between the paths when there are multiple paths to a LUN. In our case, we had only one path to each LUN.

DISK GROUP CONSIDERATIONS

We decided to use a larger stripe size for our data and log disk groups for both Oracle ASM and IBM JFS2.

Oracle ASM for 11g allows users to specify the ASM Allocation Unit (AU SIZE), while Oracle10g restricts the ASM AU SIZE to 1MB. Our testing shows that using an AU SIZE of 64MB provides higher throughput than the default size of 1MB.

To maintain an apples-to-apples comparison, we also used a 64MB stripe size in creating the JFS2 data and log disk groups.

HBA AND DISK MODIFICATIONS

`num_cmd_elems 1024`

This is the maximum number of commands sent to the HBA adapter at a given time. We used this setting to increase the number of commands that were available to the HBA.

`max_xfer_size 0x100000`

We set the maximum transfer size on the HBA adapter to match the increased maximum commands value.

queue_depth 256

Each hdisk has a queue depth setting, which allows up to 256 I/Os to be in flight against each hdisk. Because there was no disk bottleneck, we increased this value to its highest setting in order to increase the amount of I/O being sent at any given point.

max_transfer 0x100000

This is the maximum size of any given transfer for a given hdisk. We set this value for smaller I/O requests to be combined and sent in larger I/O requests. This helps optimize the I/O being sent to the storage controller.

AIX KERNEL TUNING FOR FCP

There are no special AIX kernel tuning parameters except for those shown in appendix 5.4, "AIX Kernel Parameters."

FCP CONFIGURATION

We installed NetApp FCP AIX Host Utilities 5.0. These utilities ease the setup and management of FCP on AIX with NetApp.

STORAGE CONFIGURATION

Data ONTAP 8.0 7-Mode was used for the FCP tests. The storage layout is detailed in the block protocol storage layout diagram in appendix 5.2, "Storage Layouts."

4 CONCLUSION

NetApp continues to provide leading-edge storage systems that support the NFS (kNFS and Oracle DNFS) and FCP protocols for Oracle deployments. This range of protocol support gives our customers the flexibility to choose which protocol is right for their IT infrastructure and budget. All of the protocols tested performed excellently and are suitable for production deployments. When deploying any form of NFS, you should strongly consider jumbo frames and 10GbE. Oracle DNFS and IBM JFS2 were the two top-performing test cases in this technical report, which focuses on OLTP workloads. Oracle DNFS with jumbo frames and 10GbE is an excellent alternative to FCP for customers deploying OLTP applications.

5 APPENDIXES

5.1 HARDWARE

Table 3) AIX host.

Component	Details (in Each Storage Controller)
System Type	IBM P520
Operating System	AIX 6.1-04
Processor	1 quad core CPU @ 3GHz
Memory	16GB
Oracle Database Version	11.2.0.1.0
NetApp Host Attach Kit	5.0

Component	Details (in Each Storage Controller)
FCP HBA	1 single port Emulex 4GB HBA (PCI-X)
Network Devices	1 single port 10GbE card (IBM rated for private network) 1 on-board GbE interface (public interface)

Table 4) NetApp controllers.

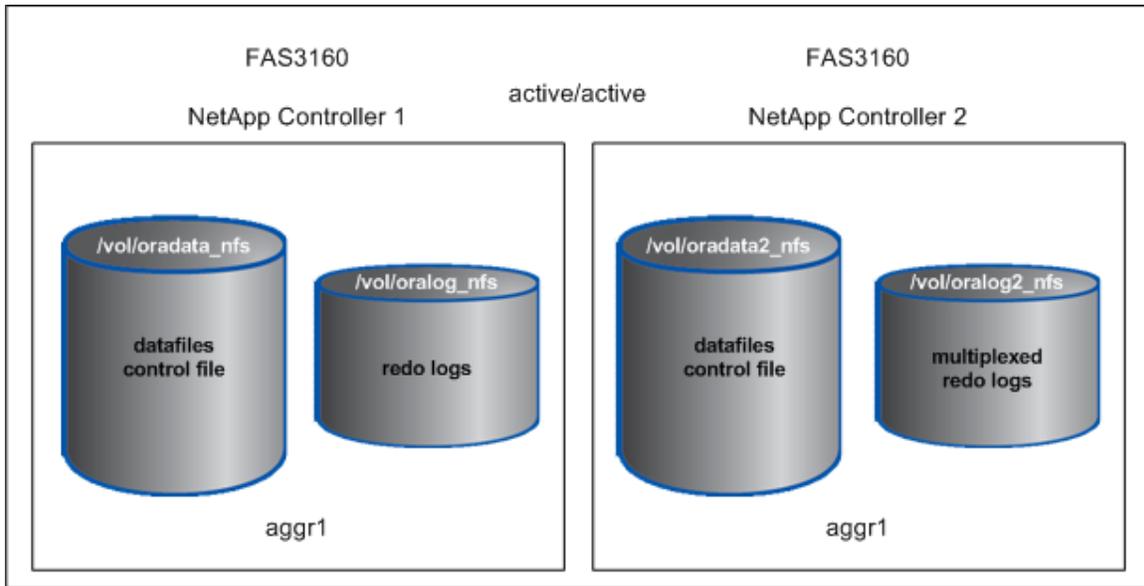
Component	Details (in each storage controller)
System Type	NetApp FAS3160 clustered
Operating System	Data ONTAP 8.0 7-Mode
Processor	1 dual core AMD CPU @ 2.6GHz
Memory	8GB cache
NVRAM	2GB
Disks	6 shelves of 300GB 15k RPM FC disks
FCP HBA	1 single port Emulex 4GB HBA (PCI-X)
Network Devices	1 single port Chelsio 10GbE card (for private network) 1 on-board GbE interface (public interface)

5.2 STORAGE LAYOUTS

KNFS AND DNFS LAYOUTS

Figure 7 shows the layout used for both kNFS and DNFS. The Oracle data files are distributed evenly across both NetApp storage controllers in the `oradata_nfs` and `oradata2_nfs` volumes. The Oracle online redo logs are also balanced across the two controllers. The primary redo log members are stored on NetApp controller 1 and all multiplexed redo log members are stored on NetApp controller 2.

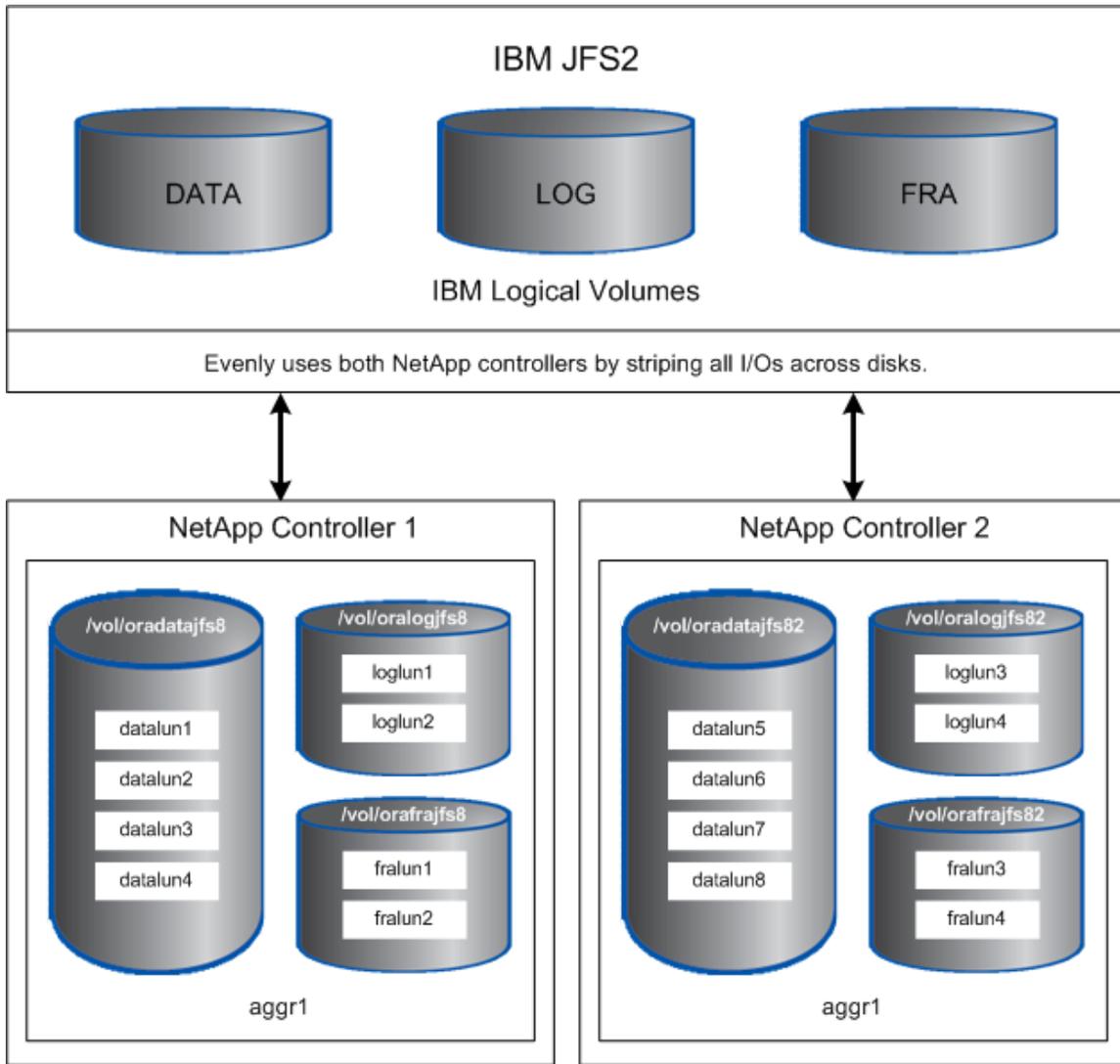
Figure 7) kNFS and DNFS volume layout.



BLOCK PROTOCOLS: JFS2 LAYOUT

Figure 8 shows that for JFS2, the Oracle data files are stored in the JFSDATA8_LV logical volume. Logical volume JFSDATA8_LV is composed of the eight total LUNs located in the `oradatajfs8` and `oradatajfs82` volumes. The Oracle online redo logs are stored in the JFSLOG8_LV logical volume, which is composed of the four total LUNs located in the `oralogjfs8` and `oralogjfs82` volumes. Multiplexed online redo logs are stored in the JFSFRA8_LV logical volume, which is composed of the four total LUNs located in the `orafrajfs8` and `orafrajfs82` volumes.

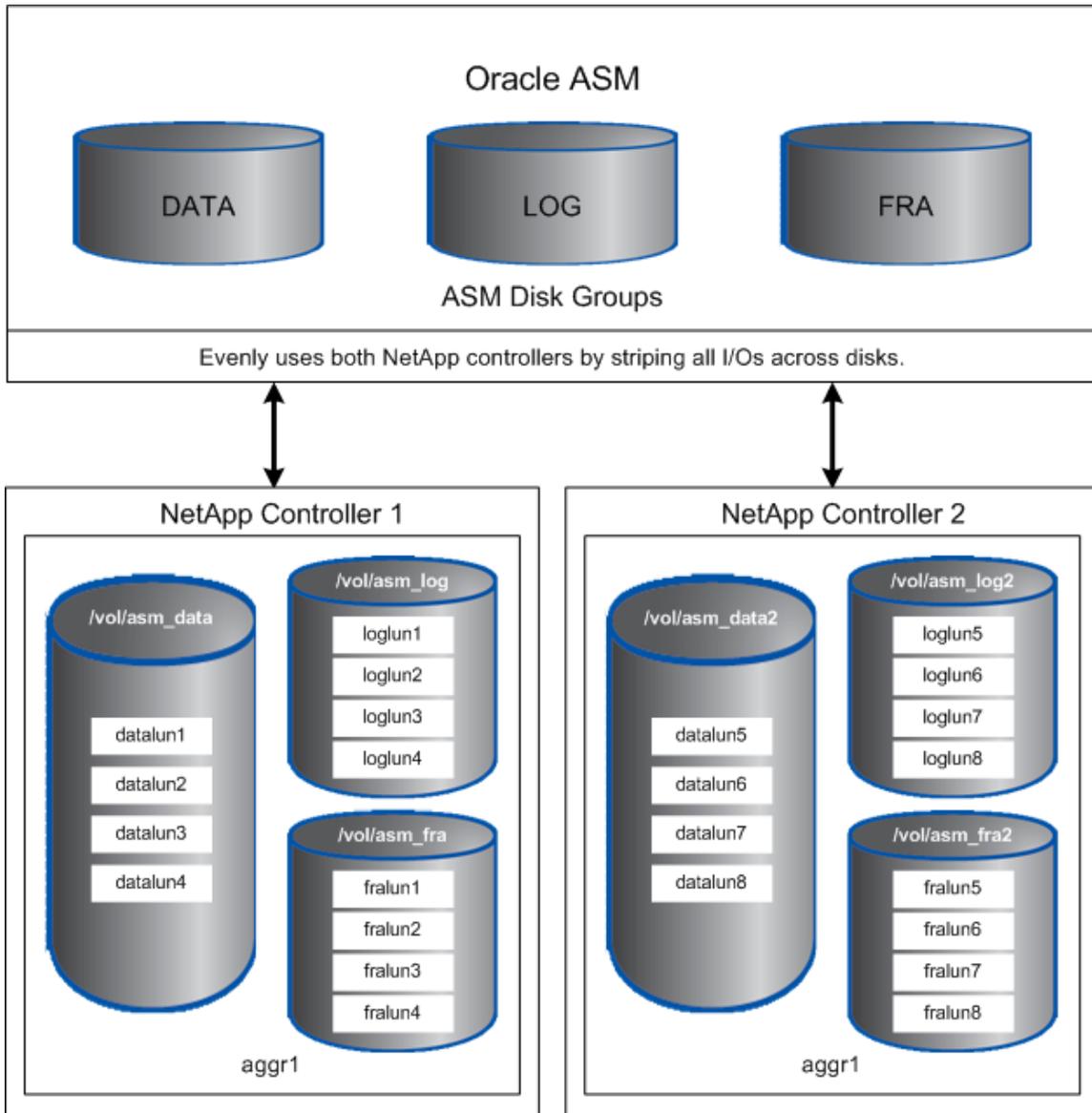
Figure 8) JFS2 volume layout.



BLOCK PROTOCOLS: ASM LAYOUT

Figure 9 shows that at the database level, the Oracle data files are stored in the ASM DATADG disk group. The DATADG disk group is composed of the eight data LUNs located in the `asm_data` and `asm_data2` volumes. The Oracle online redo logs are stored in the ASM LOGDG disk group, which is composed of the eight total LUNs located in the `asm_log` and `asm_log2` volumes. Multiplexed online redo logs are stored in the FRADG ASM disk group, which is composed of the eight total LUNs located in the `asm_fra` and `asm_fra2` volumes.

Figure 9) ASM volume layout.



5.3 ORACLE INITIALIZATION PARAMETERS

Table 5) Oracle initialization parameters (init.ora).

Parameter Name	Value	Description
<code>_in_memory_undo</code>	FALSE	Make in-memory undo for top-level transactions.
<code>_undo_autotune</code>	FALSE	Enable autotuning of <code>undo_retention</code> .
<code>compatible</code>	11.2.0.0.0	Database is completely compatible with this software version.

Parameter Name	Value	Description
cursor_space_for_time	TRUE	Use more memory in order to get faster execution.
db_block_size	8192	Size of database block in bytes.
db_cache_size	5251268608	Size of DEFAULT buffer pool for standard block-size buffers.
db_file_multiblock_read_count	128	Database block to be read each I/O.
db_files	123	Maximum allowable number of database files.
db_name	Tpcc	Database name specified in CREATE DATABASE.
disk_asynch_io	TRUE	Use asynch I/O for random access devices.
dml_locks	500	One dml locks for each table modified in a transaction.
filesystemio_options	setall	I/O operations on file system files.
log_buffer	1163264	Redo circular buffer size.
parallel_max_servers	100	Maximum parallel query servers per instance.
plsql_optimize_level	2	PL/SQL optimize level.
processes	350	User processes.
recovery_parallelism	40	Number of server processes to use for parallel recovery.
sessions	552	User and system sessions.
shared_pool_size	402653184	Size in bytes of shared pool.
statistics_level	typical	Statistics level.
transactions	350	Maximum number of concurrent active transactions.
undo_management	AUTO	If TRUE, instance runs in SMU mode; otherwise in RBU mode.
undo_retention	10800	Undo retention in seconds.
undo_tablespace	undo_1	Use or switch undo tablespace.

5.4 AIX KERNEL PARAMETERS

Table 6) AIX nondefault kernel parameters.

Parameter Name	Value	Description
maxuproc	16384	The maximum number of processes allowed per user.
maxbuf	1000	The maximum number of pages in the block-I/O buffer cache.
aio_maxservers	512	The maximum number of kernel processes per processor dedicated to AIO processing. (posix_aio_maxservers is the same as this setting. The application decides which setting to use.)
posix_aio_maxservers	512	The maximum number of kernel processes per processor dedicated to AIO processing. (aio_maxservers is the same as this setting. The application decides which setting to use.)
j2_nBufferPerPageDevice	6200	Specifies the minimum number of file system bufstructs for JFS2.
j2_dynamicBufferPreallocation	32	Specifies the number of 16k slabs to preallocate when the file system is running low of bufstructs.
j2_maxUsableMaxTransfer	1024	Specifies the maximum Logical Track Group (LTG) size, in pages, that JFS2 gathers into a single bufstruct.
j2_maxPageReadAhead	256	Specifies the maximum number of pages to be read ahead when processing a sequentially accessed file on JFS2.
numfsbufs	1024	Specifies the number of file system bufstructs.
maxpgahead	32	Specifies the maximum number of pages that are read ahead, regardless of the number of preceding sequential reads.

Parameter Name	Value	Description
minpgahead	8	Specifies the number of pages to be read ahead when the process appears to be reading sequentially from a file.
pv_min_pbuf	2048	Specifies the minimum number of pbufs per PV that the LVM uses. This is a global value that applies to all VGs on the system.
maxclient%	90	Specifies the maximum percentage of RAM that can be used for caching client pages.
maxperm%	90	Specifies the point above which the page-stealing algorithm steals only file pages.

5.5 OTHER AIX OS SETTINGS

Table 7) AIX shell limits for Oracle.

Configuration File	Settings
/etc/security/limits	fsize = -1 core = -1 cpu = -1 data = -1 rss = -1 stack = -1 nofiles = 2000

6 ACKNOWLEDGMENTS

Special thanks to the following people for their contributions: John Elliott, Saad Jafri, Chris Lemmons, and Chad Morgenstern, NetApp.

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

© Copyright 2010 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Oracle is a registered trademark of Oracle Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-3871-1010