

Linux® (RHEL 4) 64-Bit Performance with NFS, iSCSI, and FCP Using an Oracle® Database on NetApp Storage

Sanjay Gulabani, Network Appliance
Oct 2006 | TR-3495

Abstract

This technical report provides tuning recommendations that can increase the performance of Linux 2.6-based kernel environments such as Red Hat® Enterprise Linux 4 update 4 with Oracle10g™ databases. A performance comparison of NFS, iSCSI, and FCP protocols is provided. The focus of this paper is technical, and the reader should be experienced with Linux system administration, Oracle10g database administration, network connectivity, Fibre Channel administration, and NetApp storage administration.

Table of Contents

Abstract	1
1 Introduction and Summary	3
2 Hardware and Software Environment	3
2.1 Server	3
2.2 Storage	4
2.3 Kernel Parameters	4
3 Linux NFS configuration and Tuning	4
4 Linux iSCSI Configuration and Tuning	6
5 Linux FCP Configuration and Tuning	6
6 Ext3 File System Configuration	6
7 Storage System Configuration	7
8 Oracle Tuning	8
9 Oracle Performance Comparison between NFS, iSCSI, and FCP	10
10 Conclusions	14
11 Acknowledgements	14

1 Introduction and Summary

This technical report provides performance-tuning recommendations and performance comparisons for running Oracle databases over NFS, FCP, and iSCSI on Network Appliance™ storage systems in a Red Hat Enterprise Linux 4 update 4 environment. An online transaction processing (OLTP) workload using an Oracle 10.2 database is used for this comparison.

This report demonstrates that the Linux 2.6 kernel, specifically RHEL 4 update 4, has made significant improvements in NFS and iSCSI performance by improving scalability. On the hardware used for this report, NFS performance improved approximately 160% with tuning and iSCSI performance improved approximately 125% as compared to the default settings.

These measured improvements are due to software changes that allow more concurrency of I/O operations. The improvements are not out of the box; you must follow the tuning recommendations in this report. The tuning recommendations are easy to follow and do not require the painful re-layout of data files that was needed in the past to achieve good performance.

2 Hardware and Software Environment

2.1 Server

For purposes of comparison, a Sun® V40z server is used. The server configuration is shown in Table 1.

Table 1) Server configuration.

Component	Details	
Operating System	Red Hat Enterprise Linux	
Version	RHEL 4 update 4 (partner beta) 64bit	
System Type	Sun v40z	
Database Server	Oracle 10.2.0.1.0	
Total Physical RAM	16GB	
Processor	4 x Dual Core AMD Opteron™ Processor 875, 2.2GHz, 1MB cache	
Storage Network	3 x 1Gb Ethernet for NFS/iSCSI	3 x 2Gb FC-AL for FCP

2.2 Storage

The Network Appliance storage system configuration used is described in Table 2.

Table 2) Filer configuration.

Component	Details	
Operating System	Data ONTAP® 7.0.2	
Storage Interconnect	3x 1GbE for NFS and iSCSI	3x 2Gb FC-AL for FCP
Disks	18 DS14s of 144GB, 15K RPM disks (6 shelves on each controller)	
Storage Controllers	3 x FAS960	
DS to Filer	2 backside FCAL	
Storage Switches	Database server and storage system were direct connected with crossover cables	

2.3 Kernel Parameters

When running Oracle workloads of the type in this report, the following system-wide limits must be changed from the defaults:

Change system-wide limits in `/etc/sysctl.conf`

```
kernel.shmmax=12884901888
```

```
kernel.shmall=12884901888
```

3 Linux NFS Configuration and Tuning

The following mount options are used for the volumes in the RHEL 4 NFS tests:

```
rw,bg,hard,nointr,proto=tcp,vers=3,rsz=32768,wsz=32768,timeo=600
```

In Linux 2.6 kernel, the NFS module introduces a new tunable parameter, `sunrpc.tcp_slot_table_entries`. This parameter increases the concurrent I/Os to be submitted to the storage system from the default value of 16. The maximum allowable setting of `sunrpc.tcp_slot_table_entries` is 128 in the kernel used for this testing.

The OLTP performance with the default value and with `sunrpc.tcp_slot_table_entries=128` is illustrated in this report.

This parameter is designed to be set as a `sysctl` parameter; however, `sunrpc.tcp_slot_table_entries` is ignored, even when set in `/etc/sysctl.conf`, due to a Linux bug (bugzilla tracker 189310).

One workaround involves adding the following line as first line in the `/etc/init.d/netfs` file:

```
/sbin/sysctl -p
```

Adding the `/sbin/sysctl -p` command to the `netfs` script ensures that `sysctl.conf` is executed just before mounting the NFS file systems.

An alternative workaround is to unmount all NFS file systems, issue the following command, and then remount the NFS file systems:

```
#sysctl -w sunrpc.tcp_slot_table_entries=128
```

4 Linux iSCSI Configuration and Tuning

The NetApp Linux Host Utilities can be downloaded from the [NOW™](#) (NetApp on the Web) site.

The Host Utilities offer a `sanlun` utility for monitoring and mapping `/dev/sdX` devices on Linux to LUNs on the storage system.

The Linux 2.6 iSCSI kernel driver has a default `queue_depth` setting of 32 and maximum available value per device of 256. The tests demonstrate the performance obtained with the default value and the tuned value.

The `queue_depth` setting can be changed by adding the following entry to `/etc/modprobe.conf`:

```
options iscsi_sfnet can_queue=1024 cmds_per_lun=256
```

Correspondingly, the following command executed on the NetApp storage system sets the maximum queue depth on the iSCSI target:

```
#options iscsi.iswt.max_ios_per_session 256
```

For iSCSI, the traffic needs to flow on the private storage GbE only, not on the public network interface. One method to achieve that is to disable the iSCSI data traffic on the `e0` interface (or whichever interface is the public network interface in the configuration) by using one of the following commands:

```
#iswt interface disable e0 (for Data ONTAP 7.0 and earlier)
#iscsi interface disable e0 (for Data ONTAP 7.1 and later)
```

5 Linux FCP Configuration and Tuning

All of the previous configuration recommendations for iSCSI also apply to FCP devices. Additionally, the QLogic® FC-AL card has a parameter setting for `ql2xmaxqdepth`. This setting defaults to 16, and the maximum setting is 256. QLogic documentation and the SANSurfer software allow this setting to be changed. FCP tests in this report demonstrate performance with the default value of 16 and also the maximum value of 256.

6 Ext3 File System Configuration

A single LUN on each of the three storage systems is created with size 1000GB.

Each of these LUNs must be partitioned before placing a file system on it. Utilities such as `sfdisk` and `fdisk` can be used to partition the LUN.

On Linux, when a LUN is partitioned, the underlying geometry of the LUN may be reported in such a way that creating partitions on the LUN induces unaligned I/Os in the NetApp storage system. Using partitions on Linux requires that the partitions be aligned with the storage system, so that a 4096-byte

I/O on the partition results in an aligned 4096-byte single block I/O on the storage system. NetApp support knowledgebase article kb8190 illustrates very clearly how to achieve this alignment. The instructions must be followed carefully to get good performance with an ext3 file system on a partition. This article can be accessed at <http://now.netapp.com/Knowledgebase/solutionarea.asp?id=kb8190>.

For Oracle datafiles, an ext3 file system is created on top of each partition for iSCSI as well for FCP tests.

The following commands are used to create ext3 file systems and place a label on them:

```
# mke2fs -j -b 4096 /dev/sdb1
# tune2fs -L data01 /dev/sdb1
# mount -L data01 /data01
```

7 Storage System Configuration

No special tuning is recommended on the storage system. All the disks from six shelves were placed in an aggregate. The aggregate is created using RAID-DP™ with 28 disks in a RAID group. A FlexVol™ volume is created inside this aggregate.

8 Oracle Parameters

Following is a list of Oracle initialization parameters that were used for all of the OLTP tests. These are not recommended for any Oracle workload, but were used after extensive testing with our OLTP workload on our host configuration. It is neither recommended nor expected that end customers would run databases with the following parameters; they are documented for completeness.

```
compatible = 10.1.0.0.0
db_name = tpcc
control_files = (/data02/control_001, /data02/control_002)
parallel_max_servers = 32
parallel_automatic_tuning=true
recovery_parallelism= 10
db_files= 116
db_cache_size = 8000M
db_8k_cache_size= 300M
db_16k_cache_size= 900M
dml_locks= 5000
statistics_level = basic
log_buffer = 104857600
processes = 1000
sessions = 1000
transactions = 1000
shared_pool_size = 750M
cursor_space_for_time = TRUE
db_block_size = 4096
undo_management = auto
undo_retention = 2
_in_memory_undo=false
_undo_autotune=false
plsql_optimize_level=2
disk_asynch_io=true
db_writer_processes=8
```



```
db_file_multiblock_read_count=16
UNDO_TABLESPACE = undo_1
db_2k_cache_size = 20M
background_dump_dest=?/admin/bdump
core_dump_dest=?/admin/cdump
user_dump_dest=?/admin/udump
fast_start_mttr_target=2000
filesystemio_options=setall
query_rewrite_enabled=false
_optimizer_cost_model=io
pga_aggregate_target=0
_cursor_cache_frame_bind_memory=true
```

Note: RHEL 4 update 3 and later support async I/O with direct I/O on NFS. Also, `filesystemio_options = setall` was used for all protocols.

Async I/O is now recommended on all the storage protocols. The recommended value for `db_writers_processes` is to at least match the number of processor cores on the system.

9 Oracle Performance Comparison between NFS, iSCSI, and FCP

The OLTP workload Order Entry Benchmark is a series of small (typically 4KB or 8KB in size) I/Os that are a mixture of reads and writes (approximately 2:1 reads to writes). The data set is a small number of large files.

The benchmark was run in server-only mode. This means that all the users and the database engine were running on the Sun V40z.

Server-only mode also implies that the user processes are running without any think times. This means that the users continually submit transactions without simulating any delay between transactions.

Typical OLTP users have some amount of think time between keystrokes as they process transactions.

The following key metric was defined for comparisons.

OETs (order entry transactions): A measure of OETs per specified constant set of time (in this case, 1 minute).

Figure 1 shows OETs for three protocols (NFS, iSCSI, and FCP). The chart contains both default (untuned) and tuned configuration on each protocol. Note that RHEL 4 update 4* in this figure and other figures that follow is the partner-beta release of RHEL 4 update 4.

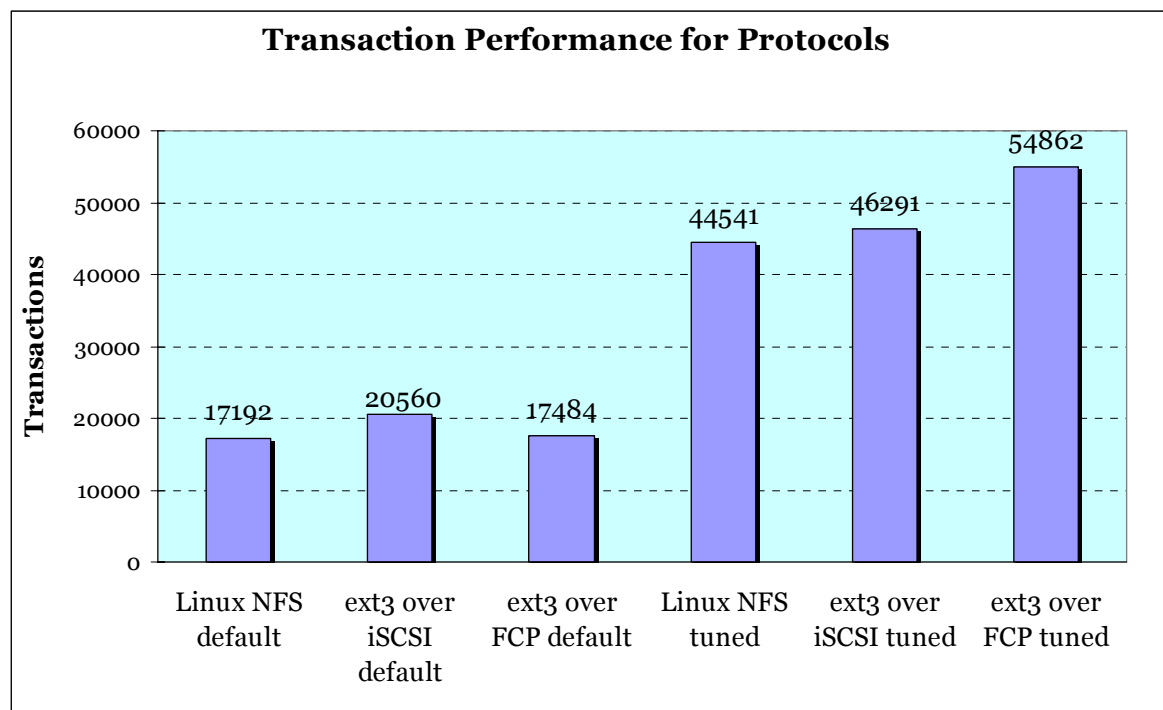


Figure 1) Order entry transactions for untuned and tuned NFS, iSCSI, and FCP.

It is observed that NFS performance improves significantly due to the ability to increase the maximum outstanding I/Os per mount point to 128 from the default of 16. The Linux 2.4 kernel supported only the fixed maximum of 16 outstanding I/Os. The Linux 2.6 kernel clearly improves this situation; however, the default value of maximum outstanding I/O is still 16.

The Linux 2.6 iSCSI kernel driver has made significant concurrency improvements as well. The default queue_depths value on Linux 2.4 was hard-coded to 12 in the iSCSI software initiator. With the Linux 2.6 kernel driver, the queue_depths value is configurable per device as well as per system. The maximum allowable limit per device is 256 and per system is 2048.

Notice that host CPU utilization for the untuned configuration (any protocol) is shown at less than 50%. This value does not increase regardless of how many additional users are added. This is a direct result of limited concurrency offered by the default settings. The performance tuning documented here clearly improves that situation.

NFS and iSCSI performance are comparable. FCP shows an improvement of 18% in transactions per minute over iSCSI.

Figure 2 shows the host CPU utilization percentage in relation to the OETs completed.

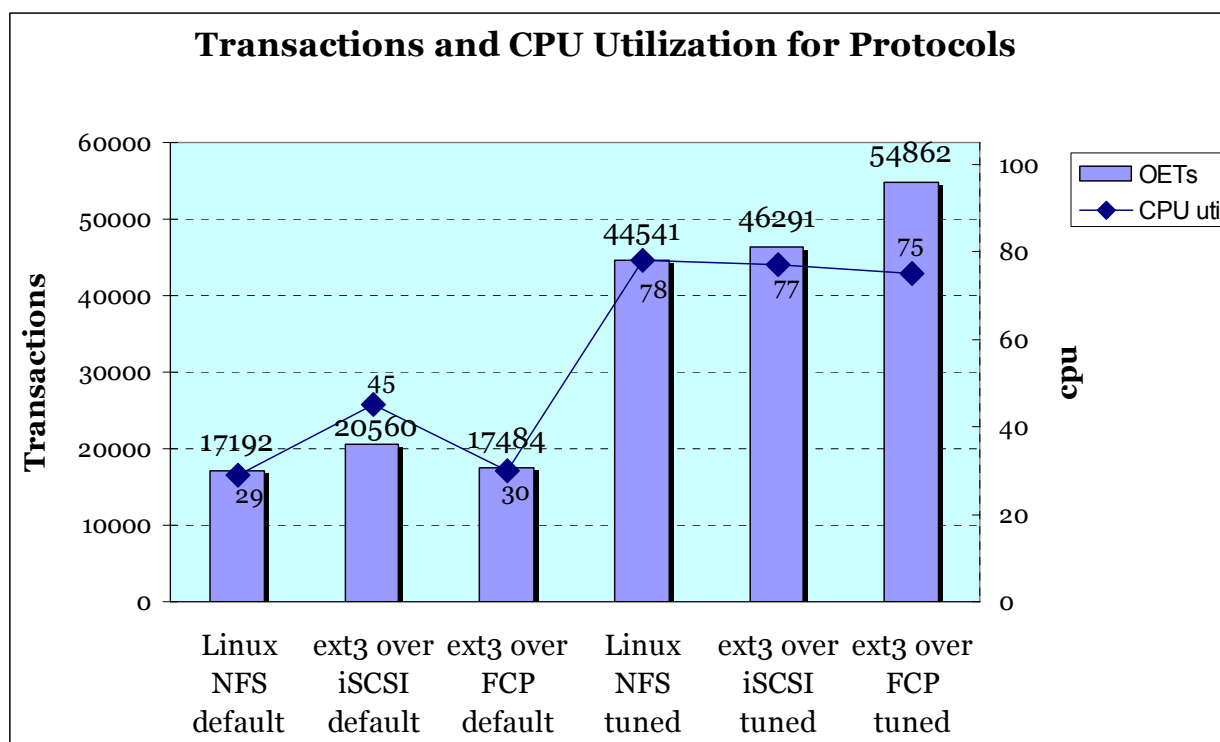


Figure 2) CPU utilization percentage in relation to OETs completed for NFS, iSCSI, and FCP.

Note that NFS and iSCSI have similar CPU costs, with an edge to NFS on lower-scale workloads. As expected, FCP has lower CPU costs than either NFS or iSCSI.

For purposes of efficiency evaluation, Figure 3 shows CPU cost per fixed number of transactions.

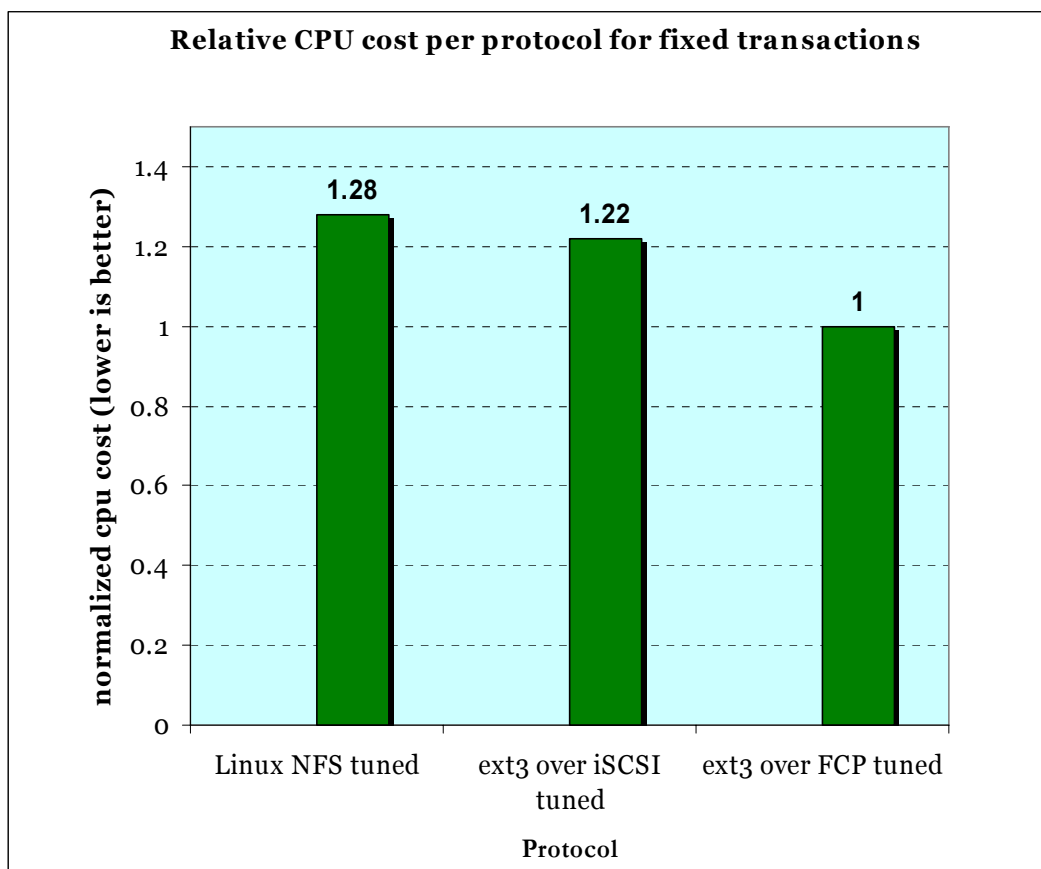


Figure 3) CPU cost for fixed transaction, NFS, iSCSI, and FCP.

The CPU costs shown in Figure 3 are calculated using $(\%CPU \text{ used} / OETs) * K$, where K is a constant such that FCP CPU cost per K transactions is 1.0 and relative cost for iSCSI and FCP is computed using the same constant K transactions.

Figure 3 shows an improvement of approximately 6% CPU efficiency going from NFS on RHEL 4 update 4 to iSCSI, and approximately 22% additional CPU efficiency gains by going to FCP in comparison with iSCSI. Also note that in the case of FCP a hardware card is used that offloads processing, and there is a dollar cost associated with that processor, which has been left out of the comparison.

The block read response times for various protocols obtained from Oracle Statspack are shown in Figure 4.

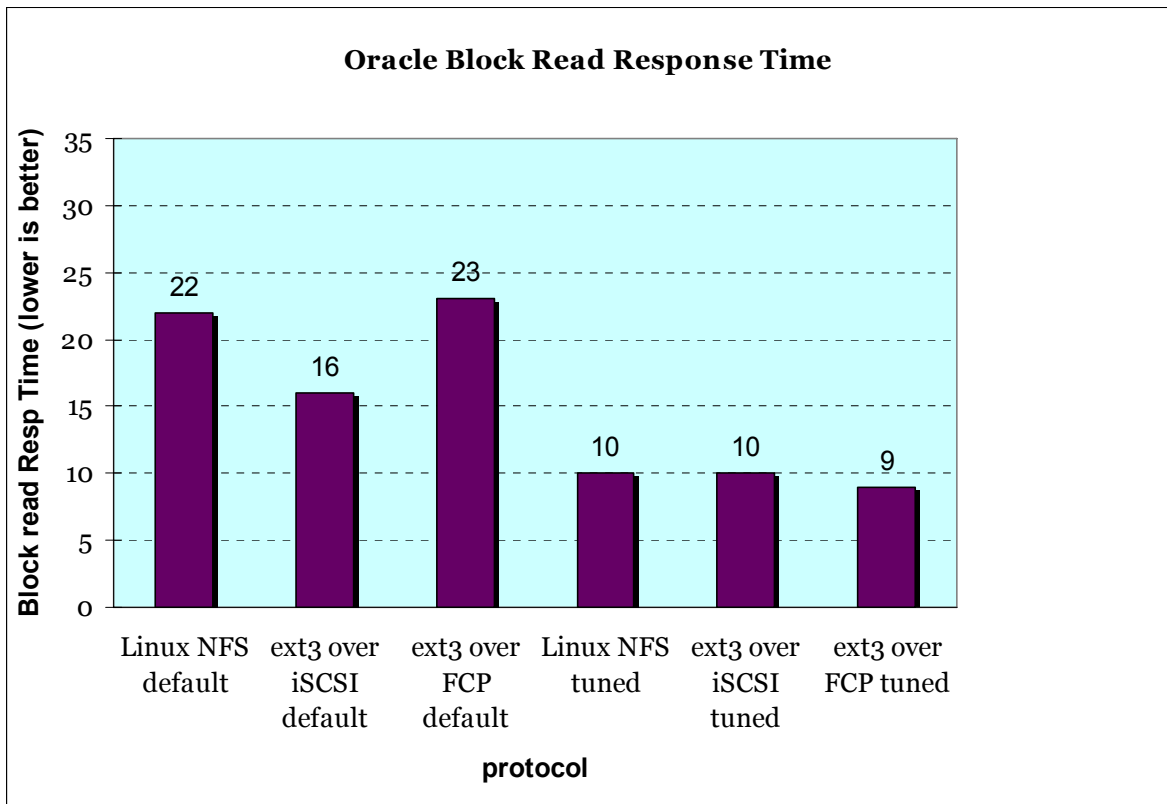


Figure 4) Oracle block read response time for untuned and tuned NFS, iSCSI, and FCP.

Note that the response times or random read latencies for all protocols with well-tuned configurations in RHEL 4 are very similar.

10 Conclusions

This paper demonstrates the concurrency improvements on NFS and iSCSI for RHEL 4 environments. It also outlines clear configuration guidelines for Oracle database workloads on all protocols that significantly improve performance and scaling over the default settings.

As with any environment, tuning a particular workload is an art. This paper suggests methods used in a lab environment that should give you good results. Individual results will vary depending on the type of the workload.

Please contact the author with any questions or comments about this document.

11 Acknowledgements

Special thanks to following people for their contributions:

Network Appliance, Inc.

Steve Daniel, Jeff Kimmel, Darrell Suggs

Red Hat, Inc.

David Wysochanski

This document will be available at Network Appliance Tech Library <http://www.netapp.com/library/tr/>.

© 2006 Network Appliance, Inc. All rights reserved. NetApp Proprietary. Specifications are subject to change without notice. NetApp, the Network Appliance logo, and Data ONTAP are registered trademarks and Network Appliance, FlexVol, NOW, and RAID-DP are trademarks of Network Appliance, Inc. in the U.S. and other countries. Linux is a registered trademark of Linus Torvalds. Oracle is a registered trademark and Oracle10g is a trademark of Oracle Corporation. Sun is a trademark of Sun Microsystems, Inc. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.