Technical Report

NetApp Best Practice Guidelines for Oracle Database 11*g*

Oracle Alliance Engineering Team, NetApp

August 2011 | TR-3633

TABLE OF CONTENTS

LIST OF TABLES

1 INTRODUCTION

Thousands of NetApp customers have successfully deployed Oracle® Databases on NetApp® storage devices for their mission-critical and business-critical applications. NetApp and Oracle have worked over the past several years to validate Oracle products on NetApp storage devices and a range of server platforms. NetApp and Oracle support have established a joint escalations team that works hand in hand to resolve customer support issues in a timely manner. In the process, the team discovered that most escalations are due to failure to follow the best established practices when deploying Oracle Databases with NetApp storage devices.

This document covers describes best practice guidelines for running Oracle Database 11*g* databases on NetApp storage systems with system platforms such as Solaris, HP/UX, AIX, Linux®, and Windows®. This document reflects the work done by NetApp, Oracle, and NetApp engineers at various joint customer sites. This document should be treated as a starting reference point and is the bare minimum requirements for deployment of Oracle on NetApp. It is intended to be a guideline of proven methods and techniques, but is not intended to cover every possible or every supported method.

This guide assumes a basic understanding of the technology and operation of NetApp products and presents options and recommendations for planning, deployment, and operation of NetApp products to maximize their effective use.

2 ORACLE DATABASE 11G NEW FEATURES

2.1 NEW FEATURES INTRODUCTION

Oracle Database 11*g* is a new major release of the Oracle Database with an impressive list of new developments and features.

By partnering with Oracle, testing, and developing advanced storage features to complement the Oracle Database 11*g* database, NetApp remains on the cutting edge with Oracle Database 11*g*.

In this section we will look at few of the new features that are of particular interest to NetApp storage users.

2.2 DIRECT NFS

Direct NFS is Oracle’s NFS client that is embedded directly into the Oracle 11*g* database application kernel. Standard NFS client software provided by the operating system vendors is not optimized for Oracle Database file I/O access patterns. With Oracle Database 11*g*, you can configure an Oracle Database to access NFSv3-configured NAS devices directly using Oracle’s Direct NFS client, rather than using the operating system kernel NFS client. The Direct NFS client accesses files stored on the NFS server through the integrated Direct NFS client, eliminating the overhead imposed by the operating system kernel NFS. These files are also accessible using the operating system kernel NFS clients, thereby enabling seamless administration. Oracle Real Application Clusters (RAC) is supported with the Direct NFS client without requiring any special configuration. Direct NFS client automatically detects when an Oracle instance is part of an RAC configuration and accordingly optimizes the NFS mount options for RAC.

2.3 ASM

Oracle Automatic Storage Management (ASM) has become a very popular feature with Oracle Database and is now well tested and thoroughly integrated with NetApp storage. Oracle has a number of new ASM enhancements with Oracle Database 11*g*. Here are some new features of ASM.

- **ASM fast mirror resync.** A new SQL statement, ALTER DISKGROUP ... DISK ONLINE, can be executed after a failed disk has been repaired. The command first brings the disk online for writes so that no new writes are missed. Subsequently, it initiates a copy of all extents marked as stale on a disk from their redundant copies. The repair time is proportional to the number of extents that have been written to or modified since the failure.

  It should be noted that NetApp RAID-DP® and/or SyncMirror® should perform the functionality of repairing and replacing failed disks transparently to Oracle such that using this command to replace a failed disk should not be needed with NetApp storage.
- **ASM manageability enhancements.** This feature includes key management improvements that further simplify and reduce management complexity for Oracle Database environments. The improvements include disk group compatibility across software versions, disk group attributes, disk group metadata backup, improved handling of missing disks at mount time, a new mount mode for more efficient rebalance performance, and extensions to the ASMCMD utility. This collection of ASM management features simplifies and automates storage management for Oracle Databases.
- **ASMCA.** ASMCA is Oracle's new GUI and CLI tool for the management and administration of the ASM disk groups. It's an enhanced version of ASMCMD, which permitted limited activity. In the past OUI or the DBCA tool was used for creating ASM instances; however, this is now performed using ASMCA. Along with this ASMCA now integrates with automatic storage management dynamic volume manager (ADVM) and automatic storage management cluster file system (ACFS).
- **ASM rolling upgrade.** Rolling upgrade is the ability of clustered software to function when one or more of the nodes in the cluster are at different software versions. The various versions of the software can still communicate with each other and provide a single system image. The rolling upgrade capability will be available when upgrading from Oracle Database 11*g* Release 1. This feature allows independent nodes of an ASM cluster to be migrated or patched without affecting the availability of the database. Rolling upgrade provides higher uptime and graceful migration to new releases.
- **ASM scalability and performance enhancements.** This feature enables Oracle to more efficiently support very large databases by transparently increasing extent size. This increases the maximum file size that Oracle can support to 128TB. Customers can also increase the allocation unit size for a disk group in powers of 2 up to 64MB. These improvements reduce database startup time and memory requirements and allow support for larger ASM files, making it feasible to implement Oracle Databases on ASM of several hundred terabytes or even petabytes. Larger allocation units provide better sequential read performance. However, note that each disk in an ASM can scale up to 2TB. To achieve the higher limits, configure multiple disks in a disk group. For more information on ASM scalability and limits, see https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=HOWTO&id=370921.1.
- **Converting single-instance ASM to clustered ASM.** This feature provides support within Enterprise Manager to convert a nonclustered ASM database to a clustered ASM database by implicitly configuring ASM on all nodes. It also extends the single-instance to Oracle RAC conversion utility to support standby databases. Simplifying the conversion makes it easier for customers to migrate their databases and achieve the benefits of scalability and high availability provided by Oracle RAC.
- **SYSASM role for ASM administration.** Oracle Database 11*g* introduces an optional system privileges role, SYSASM, and an optional operating system privileges group, OSASM, to secure privileges to perform ASM administration tasks. Oracle recommends that you use the SYSASM role instead of the SYSDBA role for Automatic Storage Management administration, to separate Automatic Storage Management administration from database administration.

Note: You can create an operating system group for Automatic Storage Management administrator, in addition to dba and oper groups.

2.4 REAL APPLICATION TESTING

Real Application Testing is a new feature of Oracle Database 11*g* that fits extremely well with NetApp FlexVol® and FlexClone® technologies. Real Application Testing has a database replay capability that captures an actual production database workload that can be replayed later for testing purposes. Real Application Testing also comes with performance analysis tools.

Database administrators constantly face the need to perform a variety of database testing and migration activities. Whether the testing is of upgrades, new applications, new SQL code, or other system modifications, everything must be thoroughly tested before it can be put into production.

Common scenarios where database testing might be required:

- Database upgrades, patches, parameter, schema changes, and so on
- Configuration changes such as conversion from a single instance to RAC, ASM, and so on
- Storage, network, interconnect changes
- Operating system, hardware migrations, patches, upgrades, parameter changes

In the past, adequate testing has been difficult to implement, and frequently with only marginal success. DBAs and system administrators have struggled to find database load generators or benchmarks that simulate the actual production database workload. Previously, a close approximation of the production workload and environment has been difficult, costly, and very time consuming to reproduce. Real Application Testing has the capability to capture an exact production database workload and then replay the workload as needed for testing purposes.

The ideal situation is to create a NetApp Snapshot™ copy at the start of the workload capture. When the capture is complete, create a database clone from the Snapshot copy and replay the workload against the clone. This will run the exact production workload against a database clone exactly as it was when the actual production workload was originally run. The clone can be recreated and the workload replayed over and over with almost no additional setup time.

Real Application Testing database replay has the following capabilities and features:

- Workload capture
- Workload processing
- Workload replay
- Analysis and reporting
- SQL performance analyzer

For more information, see: TR 3803: Upgrading to Oracle Database 11g with NetApp SnapMirror, FlexClone, and Oracle Real Application Testing

2.5 ADVANCED COMPRESSION

In recent years databases have experienced explosive size growth. Oracle Database 11g addresses this issue with a new advanced compression feature. Advanced compression works with all data types, such as regular structured data (numbers, characters), unstructured data (documents, spreadsheets, XML, and other files), and backup data.

Advanced compression in Oracle Database 11*g* not only reduces disk space requirements for all types of data; it also improves application performance and enhances memory and network efficiency. In addition, it can be used with any type application without any application changes.

Advanced compression in Oracle Database 11*g* has the following new features:

- **OLTP table compression.** This allows structured or relational data to be compressed during all types of data manipulation operations, including regular INSERT, UPDATE, or DELETEs. This new feature leverages a sophisticated and intelligent algorithm that minimizes the compression overhead during write operations, thereby making it viable for all application workloads. Additionally, it significantly improves performance of queries by reducing disk I/O and improving memory efficiency. Previous Oracle Database releases supported compression for bulk data-loading operations commonly used for data warehousing applications. Oracle Database 11*g* OLTP table compression improves database performance with more effective use of memory for caching data and reduced I/O for table scans. With OLTP table compression, you can achieve two- to threefold compression ratios with minimal processing overhead.

- **Fast files deduplication.** Intelligent technology that eliminates duplicate copies of files stored in Oracle Database 11*g*. Besides reducing storage footprint, this feature also dramatically improves the performance of write and copy operations involving duplicate content.
- **Fast files compression.** Compresses the unstructured or file data stored within the database. Two levels of compression are available so you have a choice of higher compression by using additional system (CPU) resources.
- **Backup data compression.** The storage requirements for maintaining database backups and backup performance are directly affected by database size. To that end, advanced compression includes compression for backup data when you employ Recovery Manager (RMAN) or Oracle Data Pump for database backups.
- **Network traffic compression.** Advanced compression offers the capability to compress Oracle Data Guard (standby databases) redo data as Data Guard resolves redo gaps. This improves the efficiency of network utilization and speeds up gap resolution.

2.6 ACTIVE DATA GUARD

The Oracle Active Data Guard option enables a physical standby database to be used for read-only applications while simultaneously receiving updates from a primary database. SQL queries executed on an active standby database receive up-to-date results.

2.7 SNAPSHOT STANDBY DATABASE

Oracle Data Guard has a new type of standby database called snapshot standby database. A standby database is a transactional-consistent copy of the primary database. The snapshot standby database joins the physical standby and logical standby database types of the previous Oracle Data Guard version.

A snapshot standby database is a fully updatable standby database that is created by converting a physical standby database into a snapshot standby database. A snapshot standby continues to receive, but not apply, updates generated by the primary. However these updates are automatically applied to the standby database when the snapshot standby is converted back to a physical standby database. Primary data is protected at all times. This feature provides the combined benefits of disaster recovery and reporting and testing.

The redo data received by a snapshot standby database is not applied until the snapshot standby database is converted back into a physical standby database, after first discarding any local updates made to the snapshot standby database.

A snapshot standby database is best used in scenarios that require a temporary, updatable snapshot copy of a physical standby database. Note that because redo data received by a snapshot standby database is not applied until it is converted back into a physical standby, the time needed to perform a role transition is directly proportional to the amount of redo data that needs to be applied.

2.8 NEW GRID INFRASTRUCTURE

Oracle Database 11*g* R2 provides new and unified data storage for Oracle installations. It enables placing Oracle Cluster Registry (OCR) and voting disks in an ASM disk group.

Grid infrastructure is the collection of infrastructural components provided for Oracle Database and other software. It provides integrated Clusterware functionality for cluster connectivity, messaging, locking and cluster control. ASM is now part of grid infrastructure that also contains the Clusterware software.

Grid infrastructure is now available for single instance as well. Along with the above components it also contains Oracle Restart, which is a high-availability solution for nonclustered databases. It can monitor and restart the following components if they fail: DB Instance, Oracle Net Listener, Database Services, ASM instance, and so on.

However, if any of the services is gracefully shut down by the administrator, Restart would not start them again. Restart makes sure that the database components are started in the proper order, in accordance with component dependencies.

SCAN: Single Client Access Name

All the servers in the cluster now act upon a single host name. SCAN allows cluster to be completely transparent from the users.

2.9 ON-DEMAND SEGMENT CREATION

With the new release of 11*g*, Oracle can defer segment creation for a nonpartitioned heap organized table existing in a locally managed tablespace. In this scenario, the table segment creation is deferred until the first row is inserted.

The advantages of using this new feature are:

- A significant amount of disk space can be saved for applications that create hundreds or thousands of tables upon installation, many of which might never be populated.
- Application installation time is reduced.

Note: There is a small performance penalty when the first row is inserted, because the new segment must be created at that time.

2.10 TRANSPORTABLE DATABASE

With Oracle 11*g* customers can migrate a database running in one platform to another platform using transportable database. With this release, Oracle has supported cross-platform physical standby between Linux and Windows. This is an extension of what Oracle has been supporting on transportable tablespace. For complete supportability information, see http://support.oracle.com.

3 NETAPP SYSTEM CONFIGURATION

3.1 NETWORK SETTINGS

When configuring network interfaces for new systems, it's best to run the setup command to automatically bring up the interfaces and update the `/etc/rc` file and `/etc/hosts` file. The setup command will require a reboot to take effect.

However, if a system is in production and cannot be restarted, network interfaces can be configured with the **`ifconfig`** command. If a NIC is currently online and needs to be reconfigured, it must first be brought down. To minimize downtime on that interface, a series of commands can be entered on a single command line separated by the semicolon (;) symbol.

Example:

```
netapp>ifconfig e0 down;ifconfig e0 'hostname'-e0 mediatype auto netmask

255.255.255.0 partner e0
```

Best Practice

When configuring or reconfiguring NICs or VIFs in a cluster, it is imperative to include the appropriate partner <interface> name or VIF name in the configuration of the cluster partner’s NIC or VIF to allow fault tolerance in the event of cluster takeover. Consult your NetApp support representative for assistance. A NIC or VIF being used by a database should not be reconfigured while the database is active. Doing so can result in a database crash.

ETHERNET: GIGABIT ETHERNET, AUTONEGOTIATION, AND FULL DUPLEX

Any database using NetApp storage should utilize Gigabit Ethernet on both the NetApp storage device and the database server.

NetApp Gigabit II, III, and IV cards are designed to autonegotiate interface configurations and are able to intelligently self-configure themselves if the autonegotiation process fails. For this reason, NetApp recommends that Gigabit Ethernet links on clients, switches, and NetApp systems be left in their default autonegotiation state, unless no link is established, performance is poor, or other conditions arise that might warrant further troubleshooting.

Flow control should by default be set to “full” on the NetApp storage device in its `/etc/rc` file, by including the following entry (assuming the Ethernet interface is e5):

```
ifconfig e5 flowcontrol full
```

If the output of the **`ifstat –a`** command does not show full flow control, then the switch port will also have to be configured to support it. (The **`ifconfig`** command on the NetApp storage device will always show the requested setting; **`ifstat`** shows what flow control was actually negotiated with the switch.)

3.2 VOLUME AND AGGREGATE SETUP AND OPTIONS

DATABASES

There is currently no empirical data to suggest that splitting a database into multiple physical volumes enhances or degrades performance. Therefore, the decision on how to structure the volumes used to store a database should be driven by backup, restore, and mirroring requirements.

A single database instance should not be hosted on multiple unclustered NetApp storage devices, because a database with sections on multiple NetApp storage devices makes maintenance that requires NetApp storage device downtime—even for short periods—hard to schedule and increases the impact of downtime. If a single database instance must be spread across several separate NetApp storage devices for performance, care should be taken during planning so that the impact of NetApp storage device maintenance or backup can be minimized. Segmenting the database so the portions on a specific NetApp storage device can periodically be taken offline is recommended whenever feasible.

AGGREGATES AND FLEXVOL VOLUMES

NetApp supports pooling of a large number of disks into an aggregate and then building virtual volumes (FlexVol volumes) on top of those disks. These have many benefits for Oracle Database environments; refer to [1]. NetApp also introduced support for 64-bit aggregates with this release.

For Oracle Databases it is recommended that you pool all your disks into a single large aggregate and use FlexVol volumes for your database data files and log files, as described below. This provides the benefit of much simpler administration, particularly for growing and reducing volume sizes without affecting performance. For details on exact layout recommendations, see reference [2].

VOLUME SIZE

Starting in Data ONTAP® 8.0, FlexVol volumes are of two types: 32-bit or 64-bit, depending on the type of their containing aggregate. A 64-bit volume has a larger maximum size than a 32-bit volume.

A 32-bit volume has a maximum size of 16TB. The maximum size of a 64-bit volume is determined by the size of its containing aggregate: up to 100TB, depending on the storage system model.

Note: In both types of volumes, the maximum size for LUNs and files is 16TB.

For example, even though a deduplicated volume on a FAS3270 can grow up to 16TB, some legacy file systems do not recognize volumes larger than 2TB. Therefore, NetApp recommends verifying the system limits for your operating system before deploying the solution. While designing a solution for database environments where Oracle Automated Storage Management (ASM) is used, take into account the maximum size supported by Oracle ASM disks.

By using appropriating sized volumes, you can:

- Reduce per volume backup time
- Individually group Snapshot copies and qtrees
- Improve security and manageability through data separation
- Reduce risk from administrative mistakes, hardware failures, and so on

For more information about the maximum volume size allowed for different storage systems, see the Data ONTAP Storage Management Guide at the NetApp Support (formerly NOW®) site.

For more information on deploying Oracle with ASM, see “ASM: Scalability and Limits” ID 370921.1 on Oracle Metalink.

RECOMMENDED VOLUMES FOR ORACLE DATABASE FILES AND LOG FILES

Based on our testing, we found the following layouts adequate for most scenarios. The general recommendation is to have a single aggregate containing all the flexible volumes containing database components.

Table 1) Flexible volumes and aggregate layout.

Database binaries	Dedicated FlexVol volume	
Database config files	Dedicated FlexVol volume	Multiplex with transaction logs
Transaction log files	Dedicated FlexVol volume	Multiplex with config files
Archive logs	Dedicated FlexVol volume	Use SnapMirror®
Data files	Dedicated FlexVol volume	
Temporary data files	Dedicated FlexVol volume	Do not create Snapshot copies of this volume
Cluster-related files	Dedicated FlexVol volume	ASM can be used for Oracle 11*g* R2

ORACLE OPTIMAL FLEXIBLE ARCHITECTURE (OFA) ON NETAPP STORAGE

The Oracle OFA model helps achieve the following:

- Ease backup and recovery for Oracle data and log files by putting them in separate logical volumes
- Fast recovery from a crash to minimize downtime
- Maintain logical separation of Oracle components to ease maintenance and administration
- Works well with a multiple Oracle home (MOH) layout

For more information about Oracle OFA for RAC or non-RAC, see:

- OFA for non-RAC:
  http://download.oracle.com/docs/cd/E11882_01/install.112/e16763/appendix_ofa.htm
- For RAC, OFA for ORACLE_HOME changes as:
  http://download.oracle.com/docs/cd/E11882_01/install.112/e17214/whatsnew.htm#sthref67

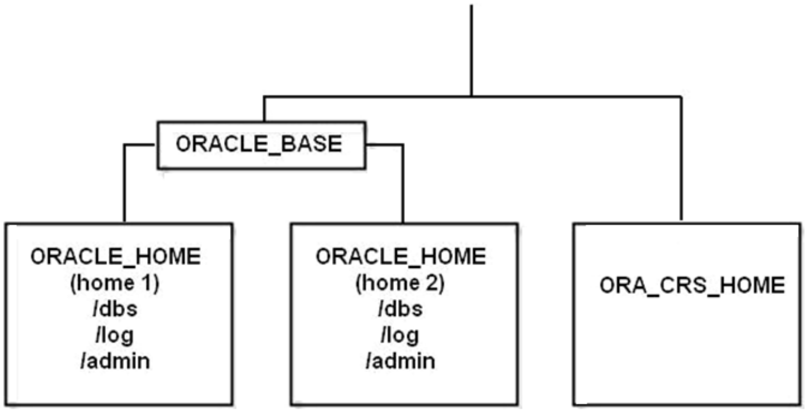

Table 2) Oracle OFA layout.

Type of Files	Description	OFA Compliant Mountpoint	Location
ORACLE_BASE		/u01/app/oracle/	Local file system or NetApp storage
ORACLE_HOME	Oracle libraries and binaries	/u01/app/oracle/11.2.0/ or /u01/app/oracle/11.2.0/db_unique_name	Local file system or NetApp storage
Data files	Oracle Database data files	/u03/oradata	NetApp storage
Log files	Oracle redo and archive logs	/u04/oradata	NetApp storage
CRS_HOME (for 10.2.x.x RAC)	Oracle CRS HOME	/u02/crs/product/10.2.0/app/	Local file system or NetApp storage
CRS_HOME (for 11.2.x.x RAC)	Oracle CRS HOME	/u01/app/crs/ (must not be a subdirectory of ORACLE_BASE)	Local file system or NetApp storage

ORACLE HOME LOCATION

OFA structure is flexible enough where ORACLE_HOME can reside either on the local file system or an NFS-mounted volume. For Oracle Database 11*g*, ORACLE_HOME can be shared for a specific RAC configuration where a single set of Oracle binaries and libraries are shared by multiple instances of the same database. Some details about shared ORACLE_HOME are discussed below.

What Is a Shared ORACLE_HOME?

- A shared `ORACLE_HOME` is an `ORACLE_HOME` directory that is shared by 2 or more hosts. This is a software install directory and typically includes the Oracle binaries, libraries, network files (listener, tnsnames, and so on), oraInventory, dbs, and so on.
- A shared `ORACLE_HOME` is a term used to describe an Oracle software directory that is mounted from an NFS server, and access is provided to 2 or more hosts from the same directory path.
- An `ORACLE_HOME` directory will look similar to the following (`/u01/app/oracle/11.1.0/db_1`) according to the OFA.

What Does Oracle Support on Oracle Database 11*g*?

- Oracle Database 11*g* supports single instance using an NFS-mounted `ORACLE_HOME` to a single host.
- Oracle Database 11*g* supports RAC using an NFS-mounted `ORACLE_HOME` to 1 or more hosts.

What Are the Advantages of Sharing the ORACLE_HOME in Oracle Database 11*g*?

- Redundant copies are not needed for multiple hosts. This is extremely efficient in a testing type of environment where quick access to the Oracle binaries from a similar host system is necessary.
- Disk space savings.
- Patch application for multiple systems can be completed more rapidly. For example, if testing 10 systems that you want to all run the exact same Oracle DB versions, this is beneficial.
- It is easier to add nodes.

What Are the Disadvantages of Sharing the ORACLE_HOME in Oracle Database 11*g*?

- When one `ORACLE_HOME` directory is patched, all databases using the same home would need to be bounced as well.
- Having a shared `ORACLE_HOME` could cause downtime to a greater number of servers if affected.

What Does NetApp Support Regarding Sharing the ORACLE_HOME?

- We *do* support a shared `ORACLE_HOME` in an RAC environment.
- We *do* support a shared `ORACLE_HOME` for single instance Oracle when mounted to a single host system.
- We *do not* support using a shared `ORACLE_HOME` in a production environment that requires high availability for a single instance Oracle setup. In other words, multiple databases should not share a single NFS-mounted `ORACLE_HOME` while any of the database are running in production mode.

BEST PRACTICES FOR CONTROL AND LOG FILES

Online Redo Log Files

Multiplex your log files. To do that, follow these recommendations:

1. Create a minimum of two online redo log groups, each with two members. Put the first member of each online redo log group on one volume and the next on another volume. The log writer (LGWR) instance process flushes the REDO log buffer, which contains both committed and uncommitted transactions to all members of the current online redo log group, and when the group is full it performs a log switch to the next group, and LGWR writes to all members of that group until the group fills up, and so on. Checkpoints do not cause log switches; in fact, many checkpoints can occur while a log group is being filled.

Archived Log Files

Set your init parameter, LOG_ARCHIVE_DEST, to a directory in the log volume such as `/u3/log/ArchiveLog` (on NetApp storage device volume `/vol/oralog3`).

Control Files

Multiplex your control files as follows:

1. Set your init parameter, CONTROL_FILES, to point to destinations on at least two different NetApp storage device volumes:

```
Dest 1: /u4/Control_File1 (on local filesystem or on NetApp storage device volume
/vol/oralog)
Dest 2: /u5/log/Control_File2 (on NetApp storage device volume /vol/oradata)
```

3.3 RAID GROUP SIZE

Configuring an optimum RAID group size for an aggregate requires a trade-off of various factors such as speed of recovery, assurance against data loss, or maximizing data storage space. In most database deployments the default RAID group size is the best size for your RAID groups. However, you can change the size of your RAID groups up to a maximum supported size.

When reconstruction rate (the time required to rebuild a disk after a failure) is an important factor, smaller RAID groups should be used. Below we recommend the best RAID group sizes based on traditional NetApp RAID-DP.

Maximum and default RAID group sizes vary according to the storage system model, level of RAID group protection provided, and types of disks used in the RAID group. NetApp recommends the following default values for the RAID group size:

- The default RAID group size for Data ONTAP 8.0.1 is 14 disks for ATA or SATA and 16 for FC or SAS.
- The maximum supported RAID group size is 28 disks for FC and 20 disks for ATA or SATA.

For more details, refer to the system configuration guide located at the NetApp Support site.

Larger RAID group sizes increase the impact from disk reconstruction due to:

- Increased number of reads required
- Increased RAID resources required
- An extended period during which I/O performance is affected (reconstruction in a larger RAID group takes longer; therefore I/O performance is compromised for a longer period)

These factors will result in a larger performance impact to normal user workloads and/or slower reconstruction rates. Larger RAID groups also increase the possibility that a maintenance effort will affect the entire RAID group.

3.4 SNAPSHOT AND SNAPRESTORE

NetApp strongly recommends using Snapshot and SnapRestore® for Oracle Database backup and restore operations. Snapshot provides a point-in-time copy of the entire database in seconds without incurring any performance penalty, while SnapRestore can instantly restore an entire database to a point in time in the past.

Note: This section assumes you are not using Snapshot or SnapRestore in SnapMirror environments. If you are using SnapMirror, see the section Consolidating Backups with SnapMirror for more information.

For Snapshot copies to be effectively used with Oracle Databases, they must be coordinated with the Oracle hot backup facility. For this reason, NetApp recommends that automatic Snapshot copies be turned off on volumes that are storing data files for an Oracle Database.

To turn off automatic Snapshot copies on a volume, issue the following command:

```
vol options <volname> nosnap on
```

If you want to make the ".snapshot" directory invisible to clients, issue the following command:

```
vol options <volname> nosnapdir on
```

With automatic Snapshot copies disabled, regular Snapshot copies are created as part of the Oracle backup process when the database is in a consistent state.

For additional information on using Snapshot and SnapRestore to back up/restore an Oracle Database, see [3].

3.5 SNAP RESERVE

Snap reserve is the amount of space reserved from a volume for the use of Snapshot copies and is expressed in percentage.

Note: Snapshot copies might consume more space than allocated with snap reserve, but user files might not consume the reserved space.

To see the snap reserve size on a volume, issue this command:

```
snap reserve
```

To set the volume snap reserve size (the default is 20%), issue this command:

```
snap reserve <volume> <percentage>
```

Do not use a percent sign (%) when specifying the percentage.

The snap reserve should be adjusted to reserve slightly more space than the Snapshot copies of a volume consume at their peak. The peak Snapshot copy size can be determined by monitoring a system over a period of a few days when activity is high.

The snap reserve may be changed at any time. Don’t raise the snap reserve to a level that exceeds free space on the volume; otherwise, client machines might abruptly run out of storage space.

Best Practice
NetApp recommends that you observe the amount of snap reserve being consumed by Snapshot copies frequently. Do not allow the amount of space consumed to exceed the snap reserve. If the snap reserve is exceeded, consider increasing the percentage of the snap reserve or delete Snapshot copies until the amount of space consumed is less than 100%.

3.6 SYSTEM OPTIONS

THE MINRA OPTION

When the minra option is enabled, it minimizes the number of blocks that are prefetched for each read operation. By default, minra is turned off, and the system performs aggressive readahead on each volume. The effect of readahead on performance is dependent on the I/O characteristics of the application. If data is being accessed sequentially, as when a database performs full table and index scans, readahead will increase I/O performance. If data access is completely random, readahead should be disabled, since it might decrease performance by prefetching disk blocks that are never used, thereby wasting system resources.

The following command is used to enable minra on a volume and turn readahead off:

```
vol options <volname> minra on
```

Generally, the readahead operation is beneficial to databases, and the minra option should be left turned off. However, NetApp recommends experimenting with the minra option to observe the performance impact, as it is not always possible to determine how much of an application’s activity is sequential versus random. This option is transparent to client access and can be changed at will without disrupting client I/O. Be sure to allow two to three minutes for the cache on the appliance to adjust to the new minra setting before looking for a change in performance.

Note: Historically, NetApp had recommended disabling aggressive storage readahead for OLTP database workloads by setting the Data ONTAP parameter "`minra`" to "`on`." Data ONTAP 6.5.1, however, introduced significant changes to the readahead algorithm, making it more intelligent and efficient. As a result, NetApp no longer recommends disabling readahead for database workloads. Enabling `minra` might lower the overall database performance. As a result, NetApp now recommends that the `minra` setting be left in the default "`off`" state unless explicit guidance to do otherwise is given by the NetApp Global Support organization.

FILE ACCESS TIME UPDATE

Another option that can improve access time is file access time update. If an application does not require or depend upon maintaining accurate access times for files, this option can be disabled. Use this option only if the application generates heavy read I/O traffic. The following command is used to disable file access time updates:

```
vol options <volname> no_atime_update on
```

NFS V3 SETTINGS

NFS v3 is currently supported for all Oracle versions and implementations that support NFS. This includes Oracle single instance and RAC. Detailed NFS v3 setup and configuration information can be found in the NetApp technical reports for Oracle installation on each specific platform. NetApp supports the use of TCP as the data transport mechanism with the current NFS v3 client software on the host. UDP is not supported as the data transport mechanism for Oracle data files.

More information can also be found in the NetApp database mount options page: http://kb.netapp.com/support/index?page=content&id=3010189.

NFS V4 SETTINGS

Network File System (NFS) version 4 is a distributed file system protocol based on NFS protocol versions 2 [RFC1094] and 3 [RFC1813]. Unlike earlier versions, the NFS version 4 protocol supports traditional file access while integrating support for file locking and the mount protocol. In addition, support for strong security (and its negotiation), compound operations, client caching, and internationalization has been added. Attention has also been applied to making NFS version 4 operate well in an Internet environment.

The goals of the NFS version 4 revisions are as follows:

- Improved access and good performance on the Internet
- Strong security with negotiation built into the protocol
- Good cross-platform interoperability
- Designed for protocol extensions

The general file system model used for the NFS version 4 protocols is the same as previous versions. The server file system is hierarchical; the regular files contained within are treated as opaque byte streams. In a slight departure, file and directory names are encoded using UTF-8 to deal with the basics of internationalization. NetApp released the first NFSv4 server in Data ONTAP 6.4. NFSv4 client implementations have changed since Data ONTAP 6.4 was released, and this has introduced some interoperability issues. Data ONTAP 7.3 addresses many of these issues. Several client operating systems now support NFSv4 to varying degrees.

For more details on the NFSv4 supportability with Oracle, see the Interoperability Matrix located on NetApp Support site.

FREE SPACE MANAGEMENT AND ENOSPC

It is important that Oracle users understand the effects of free space management within NetApp, and it is their responsibility to manage that free space. In this section we cover some of the choices for effective free space management and also show how NetApp deals gracefully (no corruption) in instances where free space has not been managed effectively and the operating system has returned an ENOSPC error.

ENOSPC is a UNIX® operating system error that sometimes returns the message “Not enough space is available to service your request." The error message occurs because of a shortage of file system space or lack of available media blocks. Using Solaris as an example, this is errno 28 as defined in the header file:

```
/usr/include/sys/errno.h:
28 ENOSPC    No space left on device
```

The problem was associated with block devices earlier, which was taken care by the 2X+delta space reservation rule. The ENOSPC error can also be encountered in NFS-based implementations and is becoming more frequent with the FlexVol and FlexClone utilization when not doing proper storage planning. Even though the ENOSPC error causes a graceful crash of an Oracle instance or offlining of a tablespace, the unwanted interruptions can easily be avoided with better planning.

The error message occurs because of a shortage of file system space or lack of available media blocks. The error can result from one of the following conditions:

- The application seeks past the EOM/EOP on the device.
- The device containing the file referred to by the file descriptor has no room for the data.
- The disk quota is exhausted and will not accept additional write requests.
- The underlying file system is full, and there is no room for file metadata to expand.

These are some conditions under which an ENOSPC error might result:

- Snapshot copies consuming the volume space
- More flexible clones with high data change rates

Let's look at what happens when Oracle receives the ENOSPC error. Oracle preallocates the space needed by data containers. When Oracle receives an ENOSPC error on a precommitted space, it reacts in different ways, depending on which type of database file got the error. The recovery method depends on the type of failure. Keep in mind that the problem is not Oracle corruption.

A frequently asked question is: Why are we getting ENOSPC errors when we have precommitted space? We get ENOSPC errors because Data ONTAP lets the Snapshot copy grow into the volume space. Every write in WAFL® is a write to a new block. If an old block is part of a Snapshot copy, Data ONTAP needs to preserve the old block and the new changed block. This is not a problem specific to NetApp. Every storage vendor who supports a snapshot feature has to deal with it. There are two options when there is no space to accommodate the Snapshot copies:

- Delete old Snapshot copies as the Snapshot copies grow to reduce delta
- Preserve the older Snapshot copies and generate an error on the active file system

Data ONTAP chooses the second option. Deleting backups is far more dangerous.

Why does Oracle behave differently at different times? The Oracle engine behaves differently depending on which file causes this error. For any file that is critical to the consistency of the database (system tablespace, online redo logs), the instance crashes. Errors on user data files will offline the data file and report an error on append operations. An example of an append operation would be adding data to the data file. An error writing to the archive log will put the database in wait mode until space is available in the flexible volume for the archive log to be written to. After adding additional space to the volume, the database should continue to work normally. For details, see Table 3.

How are FlexClone flexible volumes affected? There is nothing specific to flexible volumes or flexible clones regarding this error. It is just a matter of how much of free space is left in the corresponding virtual data container.

If the ENOSPC error occurs on a user data file, the data file will be put in recovery mode and offlined. The procedure for recovery is as simple as doing an instance recovery. All we need to do is replay the online redo logs. The blocks corresponding to the previous checkpoint have already been written to the data file. Note that pulling an older copy of the data file and performing recovery is going to take a longer time and could cause cascaded crashes (getting another ENOSPC while performing recovery). Add enough space in the flexible volume before performing recovery to avoid this cascaded crash problem.

How to Avoid Getting ENOSPC Errors

It is quite rare to see these types of errors. However, to avoid seeing this interruption of your database, NetApp recommends the following steps be taken:

- Plan your storage well.
- Identify the amount of storage is needed in your environment and revisit that plan on a monthly basis.
- Utilize the autogrow feature of a FlexVol volume if you are using Data ONTAP 7.1 or greater.
- Use the 100% volume/file space reserve for your production environments.
- Make sure that enough space is available in the volume/aggregate before setting up test and development environments with FlexClone.
- Allocate enough Snapshot reserve, especially if you are anticipating large data change deltas (large amounts of data written) between Snapshot copies.
- If you have an ENOSPC failure, make sure to grow your flexible volume before doing a recovery.

Table 3) ENOSPC errors.

Types of Data Files Used for Failure Scenarios	Types of Errors Reported	After Error Received, What Was Required from Storage Perspective?	Recovery Steps Needed
A NetApp volume containing user-level data files (that is, users01.dbf, and so on)	ORA-27072 KCF: write/open error No space left on device ORA-00603: Oracle server session terminated by fatal error	1. Recover data file 2. Alter database data file '/ab/ab.dbf' online	Recover the data file that went offline. You should be able to do that online without restarting the DB, provided that you had the proper archived logs. If there are no archived logs, you can recover current redo logs. If the database is out of sync, you need to restore the previous backup.
A NetApp volume containing Oracle control files (that is, control01.ctl, and so on)	ORA-27072: File I/O error ENOSPC reported by logwriter (lgwr)	Add space and restart the database.	DB started up normally.
A NetApp volume containing Oracle transaction log files (that is, redo01.log, and so on)	ORA-03113: end-of-file on communication channel ENOSPC reported by logwriter (lgwr)	Add space to volume and restart the database.	No recovery required. - startup nomount - alter database mount - alter database open

Types of Data Files Used for Failure Scenarios	Types of Errors Reported	After Error Received, What Was Required from Storage Perspective?	Recovery Steps Needed
A NetApp volume containing system-level data files (that is, system01.dbf, and so on)	ORA-27072: File I/O error ORA-01243: system tablespace file suffered media failure. Dbwriter (dbw0) reported the ENOSPC	Add space and specify startup mount and then alter database open.	DB reported media failure but did an instance recovery on startup
A NetApp volume containing temporary database files (that is, undotbs01.dbf, and so on)	KCF: write/open error=27072 No space left on device ENOSPC reported by dbwriter (dbw0)	Recover the file and continue the operation. - Recovery data file - Alter database data file '…' online	Recovery in online mode without restarting the DB
A NetApp volume containing archive log files (that is, arc0805_xxx.dbf, and so on)	ENOSPC reported by archiver process	Once archive log volume full no more writes are written out.	Increased the volume and archiver kept working

DIRECT NFS

Direct NFS (DNFS) is a new feature provided in the Oracle Database 11*g* release. One of the primary challenges of operating system kernel NFS administration is the inconsistency in managing NFS configurations across different operating system platforms. The Direct NFS client eliminates this problem by providing a standard NFS client implementation across all platforms supported by the Oracle Database. This also makes NFS a viable solution even on platforms that don't support NFS, for example, Windows. The Oracle Direct NFS client is implemented as an Oracle Disk Manager (ODM) interface and can be easily enabled or disabled by linking the relevant libraries.

Benefits of Direct NFS Client (DNFS)

The Direct NFS client in the Oracle Database 11*g* release overcomes the variability in NFS I/O performance by hosting the NFS client software inside the database kernel and is therefore independent of the operating system kernel.

The NFS client embedded in an Oracle 11*g* database provides the following benefits:

- Stable and consistent NFS performance is observed across all operating system platforms.
- The Direct NFS client is modified to better cache and manage I/O patterns typically observed in database environments, for larger and more efficient reads and writes.
- The Direct NFS client allows asynchronous direct I/O, which is the most efficient setting for databases. This significantly improves read/write database performance by allowing I/O to continue while other requests are being submitted and processed.

- Database integrity requires immediate write access to the database when requested. Operating system caching delays write access for efficiency reasons, potentially compromising data integrity during failure scenarios. The Direct NFS client uses the database caching techniques with asynchronous direct I/O to make sure the data writes occur expeditiously, thus reducing data integrity risks.
- The NFS client manages load balancing and high availability by incorporating these features directly in the Direct NFS client, rather than depending on the operating system kernel NFS clients. This greatly simplifies network setup in high-availability environments and reduces dependence on network administrators. This eliminates the need to set up network subnets and bonded ports such as Link Aggregation Control Protocol (LACP) bonding.
- The Direct NFS client allows up to four parallel network paths/ports to be used for I/O between the database server and the NAS storage system. For efficiency and performance, these are managed and load-balanced by the Direct NFS client and do not depend on the operating system.
- DNFS overcomes operating system write-locking, which can be inadequate in some operating systems and can cause I/O performance bottlenecks in others.
- Database server CPU and memory usage are reduced by eliminating the overhead of copying data to and from the operating system memory cache to the database system global area (SGA).

Direct NFS Installation

The Direct NFS client does not require any special installation steps as it is installed automatically during the regular Oracle 11*g* database install process. It is turned off by default but can be enabled by using the following steps:

Linux or UNIX:

```
OSPROMPT> cd $ORACLE_HOME/lib
OSPROMPT> cp libodm11.so libodm11.so_orignal_stub
OSPROMPT> ln -s libnfsodm11.so libodm11.so
```

Windows:

```
OSPROMPT> cd %ORACLE_HOME%\bin
OSPROMPT> copy libodm11.dll libodm11.dll_orignal_stub
OSPROMPT> copy /Y libnfsodm11.dll libodm11.dll
```

Direct NFS Configuration

The first step to using the Direct NFS client is to make sure that all of the Oracle Database files residing on the NetApp storage system volumes are mounted using kernel NFS mounts. Direct NFS does not require any special NFS mount options. However, it needs the rsize and wsize NFS mount options to be set to 32768 (32K) as the max value of `DB_BLOCK_SIZE` can be 32K.

See the recommended NFS mount options for Oracle Databases in http://kb.netapp.com/support/index?page=content&id=3010189.

The Direct NFS client uses a new configuration file, `oranfstab`, or the mount tab file (`/etc/mtab` on Linux) to determine the mountpoint settings. The Direct NFS client needs just one configuration file whose format is almost identical across all operating systems. A brief introduction of these files is given below.

Linux or UNIX:

- **oranfstab**: `oranfstab` is a special configuration file for the Direct NFS client. This file is located in `$ORACLE_HOME/dbs/oranfstab` or `/etc/oranfstab`. The `oranfstab` file with the Direct NFS client permits the usage of Oracle Database on the Windows platform using the NFS protocol. The `oranfstab` file contains the NFS server and local interface details and mountpoints for various volumes exported by the NFS server.
- **kernel mount tab:** Kernel mount tab file is located in `/etc/mtab` on most UNIX variants. This file stores information regarding mounted data volumes on the system. The Direct NFS client defaults to

the mount tab file if it cannot identify the necessary files available through the `oranfstab` file, or if the `oranfstab` file cannot be found on the system. The Direct NFS client searches for mount entries in the following order:

1. `$ORACLE_HOME/dbs/ornfstab`
2. `/etc/oranfstab`
3. `/etc/mtab`

The Direct NFS client uses the first matching entry found.

The following example shows an `oranfstab` configuration file that maps two NFS volumes from an NFS server that can be addressed using multiple (4) names over four different paths.

Example: Sample `oranfstab` on Linux

```
server: MyDataServer1
local: LocalPath1
path: NfsPath1
local: LocalPath2
path: NfsPath2
local: LocalPath3
path: NfsPath3
dontroute
export: /vol/oradata1 mount: /mnt/oradata1
export: /vol/oradata2 mount: /mnt/oradata2
export: /vol/oradata3 mount: /mnt/oradata3
export: /vol/oradata6 mount: /mnt/oradata6
```

Where:

- `server` represents the NFS server name.
- `MyDataServer1` is the storage system name. Even though it is possible to have all paths on the same subnet, it is recommended to keep each path on its own subnet for better availability.
- `local` is the network path from the database host. Up to four local paths on the database host, specified by IP address or by name, can be used, as displayed using the **`ifconfig`** command run on the database host.
- `path` is the network interfaces on the storage system. Up to four network paths to the storage system specified either by IP address, or by name, can be used, as displayed using the **`ifconfig`** command on the storage system.
- `dontroute` specifies that outgoing messages should not be routed by the operating system, but sent using the IP address to which they are bound.
- `export` is the exported path from the NFS server.
- `mount` attribute represents the local mountpoint on the database host.

Note: The parameters `local` and `dontroute` are available from patchset 11.1.0.7 onward.

Direct NFS can use up to four network paths defined for an NFS server. The Direct NFS client performs load balancing across all specified paths. If a specified path fails, then Direct NFS reissues I/Os over any remaining paths. For more information on configuring DNFS for multiple paths, see https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&doctype=HOWTO&id=822481.1.

Windows

The Direct NFS client requires a special configuration file to map the NFS server exports to local mountpoints. The default location of this file is `%ORACLE_HOME%\dbs\oranfstab`.

The database files accessed through the Direct NFS client should also be mounted using other means, such as the Common Internet File System (CIFS) in the case of Windows. This makes sure that the kernel I/O interface is able to access these files.

The example below shows an `oranfstab` configuration file that maps three NFS volumes from an NFSServer that can be addressed using multiple (4) names over four different paths.

Example 1: Sample `oranfstab` on Windows

```
server: NFSServer
path: NFSPath1
path: NFSPath2
path: NFSPath3
path: NFSPath4
export: /vol/oradata1 mount: D:\ORACLE\ORADATA1
export: /vol/oradata2 mount: D:\ORACLE\ORADATA2
export: /vol/oralog mount: D:\ORACLE\ORALOG1
uid: xxxxx
gid: yyyyy
```

The Windows `oranfstab` configuration file contains some extra options to handle NFS security. The 'uid' and 'gid' parameters can be appended to the end of each server configuration, which represent the UNIX user ID to be used by the Direct NFS client, and gid represents the UNIX group ID to be used by the Direct NFS client.

Direct NFS Configuration for Real Application Clusters

In Oracle Database 11*g*, only the files related to Oracle such as data files, online redo logs, archive log files, and temporary files can be accessed using the Oracle DNFS client. The Oracle Clusterware (CRS)-related files cannot be accessed over the Direct NFS path and require a native NFS mount with the right mount options. See section FCP SAN Initiators for Linux for the right mount options for CRS volumes. For file storage options, see section 4.7.

For more information, see document ID 762374.1 on my Oracle Support.

Direct NFS Sizing Considerations

The Oracle Direct NFS client consistently performs better than the native NFS client and closes the performance gap between NFS and FCP for some workloads. It is recommend to size for 10% to 20% more I/O performance when sizing storage for a database application with the Direct NFS client.

Storage of Oracle Clusterware Files for DNFS

The Oracle Cluster Registry and voting files may not be stored on a DNFS mountpoint. For OCR and voting file storage options, see section 4.7.

Direct NFS Client Recommended Patches

The following Oracle patches are mandatory depending on the Oracle Database version being used:

- 11.2.0.1: Most bug fixes are available using patch 8981354.
- 11.2.0.2: Most bug fixes are available using patch 9977452.

4 ORACLE DATABASE SETTINGS

4.1 DISK_ASYNCH_IO

Enables or disables Oracle asynchronous I/O. `DISK_ASYNCH_IO` controls whether I/O to data files, control files, and log files is asynchronous (that is, whether parallel server processes can overlap I/O requests with CPU processing during table scans). If the platform supports asynchronous I/O to disk, Oracle recommends that you leave this parameter set to its default value of TRUE. However, if the asynchronous I/O implementation is not stable, you can set this parameter to false to disable asynchronous I/O. If the platform does not support asynchronous I/O to disk, this parameter has no effect.

Asynchronous I/O allows processes to proceed with the next operation without having to wait for an issued write operation to complete, therefore improving system performance by minimizing idle time. This setting might improve performance depending on the database environment. If the `DISK_ASYNCH_IO` parameter is set to TRUE, then `DB_WRITER_PROCESSES` or `DBWR_IO_SLAVES` must also be used to gain any performance advantage, as described below. The calculation is as follows:
DB_WRITER_PROCESSES = 2 * number of CPU cores.

Choosing Between Multiple DBWR Processes and I/O Slaves

Configuring multiple DBWR processes benefits performance when a single DBWR process is unable to keep up with the required workload, especially when the database server has multiple CPUs. However, before configuring multiple DBWR processes, check whether asynchronous I/O is available and configured on the system. If the system supports asynchronous I/O but it is not currently used, then enabling asynchronous I/O might provide a performance improvement. If the system does not support asynchronous I/O, or if asynchronous I/O is already configured and there is still a DBWR bottleneck, then configure multiple DBWR processes.

Note: If asynchronous I/O is not available on your platform, then asynchronous I/O can be disabled by setting the `DISK_ASYNCH_IO` initialization parameter to FALSE.

Using multiple DBWRs parallelizes the gathering and writing of buffers. Therefore, multiple DBWn processes should deliver more throughput than one DBWR process with the same number of I/O slaves. For this reason, multiple DBWR processes are preferred over the use of I/O slaves. I/O slaves should only be used if multiple DBWR processes cannot be configured.

4.2 DB_FILE_MULTIBLOCK_READ_COUNT

Determines the maximum number of database blocks read in one I/O operation during a full table scan. The number of database bytes read is calculated by multiplying DB_BLOCK_SIZE * DB_FILE_MULTIBLOCK_READ_COUNT. The setting of this parameter can reduce the number of I/O calls required for a full table scan, thus improving performance. Increasing this value might improve performance for databases that perform many full table scans but degrade performance for OLTP databases where full table scans are seldom (if ever) performed.

Setting this number to a multiple of the NFS READ/WRITE size specified in the mount will limit the amount of fragmentation that occurs in the I/O subsystem. Be aware that this parameter is specified in “DB Blocks,” and the NFS setting is in “bytes,” so adjust as required. As an example, specifying a DB_FILE_MULTIBLOCK_READ_COUNT of 4 multiplied by a DB_BLOCK_SIZE of 8kB will result in a read buffer size of 32kB.

As of Oracle Database 10*g* R2 and later, the default value of this parameter is a value that corresponds to the maximum I/O size that can be performed efficiently. This value is platform dependent and is 1MB for most platforms. Because the parameter is expressed in blocks, it will be set to a value that is equal to the maximum I/O size that can be performed efficiently divided by the standard block size. Note that if the number of sessions is extremely large, the multiblock read count value is decreased to avoid the buffer cache getting flooded with too many table scan buffers.

The maximum value is the operating system's maximum I/O size expressed as Oracle blocks ((max I/O size)/DB_BLOCK_SIZE). If you set this parameter to a value greater than the maximum, Oracle uses the maximum.

In some situations DB_FILE_MULTIBLOCK_READ_COUNT was shown to have no impact. In some recent testing DB_FILE_MULTIBLOCK_READ_COUNT was shown to have less effect on full table scan performance than degree of parallelism using Oracle parallel query.

In some cases increasing the value of DB_FILE_MULTIBLOCK_READ_COUNT might improve scan performance. Typically recommended values are 16, 32, or 64. Some testing might be required in the specific environment to optimally tune DB_FILE_MULTIBLOCK_READ_COUNT. For more information on this parameter, see references [14] and [15].

4.3 DB_BLOCK_SIZE

DB_BLOCK_SIZE specifies (in bytes) the size of Oracle Database blocks. Typical values are 4096 and 8192. The value of this parameter must be a multiple of the physical block size at the device level. The default value is 8192 for Oracle Database 11*g*. The range of values is 2048 to 32768, but some operating systems might have a narrower range.

The value for DB_BLOCK_SIZE is in effect at the time you create the database and determines the size of the blocks. The value must remain set to its initial value.

For RAC, this parameter must be set for every instance, and multiple instances must have the same value. This parameter affects the maximum value of the FREELISTS storage parameter for tables and indexes for RAC. Oracle uses one database block for each freelist group. Decision support system (DSS) and data warehouse database environments generally tend to benefit from larger block size values.

For best database performance, DB_BLOCK_SIZE should be a multiple of the OS block size. For example, if the Solaris block size is 4096:

DB_BLOCK_SIZE = 4096 * n

The NFS rsize and wsize options specified when the file system is mounted should also be a multiple of this value. Under no circumstances should they be smaller. For example, if the Oracle DB_BLOCK_SIZE is set to 16kB, the NFS read and write size parameters (rsize and wsize) should be set to either 32kB or 64kB, never to 8kB or 4kB.

4.4 DBWR_IO_SLAVES AND DB_WRITER_PROCESSES

DB_WRITER_PROCESSES is useful for systems that modify data heavily. It specifies the initial number of database writer processes for an instance. If DBWR_IO_SLAVES is used, only one database writer process will be allowed, regardless of the setting for DB_WRITER_PROCESSES. Multiple DBWRs and DBWR I/O slaves cannot coexist. It is recommended that one or the other be used to compensate for the performance loss resulting from disabling DISK_ASYNCH_IO. Metalink note 97291.1 provides guidelines on usage.

The first rule of thumb is to always enable DISK_ASYNCH_IO if it is supported on that OS platform. Next, check to see if it is supported for NFS or only for block access (FC/iSCSI). If supported for NFS, then consider enabling async I/O at the Oracle level and at the OS level and measure the performance gain. If performance is acceptable, then use async I/O for NFS. If async I/O is not supported for NFS or if the performance is not acceptable, then consider enabling multiple DBWRs and DBWR I/O slaves, as described next.

Multiple DBWRs and DBWR I/O slaves cannot coexist. It is recommended that one or the other be used to compensate for the performance loss resulting from disabling DISK_ASYNCH_IO. The recommendation is that DBWR_IO_SLAVES be used for single CPU systems and that DB_WRITER_PROCESSES be used with systems having multiple CPUs.

I/O slaves might be used to simulate asynchronous I/O on platforms that do not support asynchronous I/O or that implement it inefficiently. However, I/O slaves might also be useful even when asynchronous I/O is being used.

DBWR_IO_SLAVES is intended for scenarios where you cannot use multiple DB_WRITER_PROCESSES (for example, where you have a single CPU). I/O slaves are also useful when asynchronous I/O is not available, because the multiple I/O slaves simulate nonblocking, asynchronous requests by freeing DBWR to continue identifying blocks in the cache to be written. Asynchronous I/O at the operating system level, if you have it, is generally preferred.

DBWR I/O slaves are allocated immediately following database open when the first I/O request is made. The DBWR continues to perform all of the DBWR-related work, apart from performing I/O. I/O slaves simply perform the I/O on behalf of DBWR. The writing of the batch is parallelized between the I/O slaves.

Multiple DBWR processes cannot be used with I/O slaves. Configuring I/O slaves forces only one DBWR process to start.

Configuring multiple DBWR processes benefits performance when a single DBWR process is unable to keep up with the required workload.

Generally, multiple DBWn processes should deliver more throughput than one DBWR process with the same number of I/O slaves. I/O slaves should only be used if multiple DBWR processes cannot be configured.

Best Practice
NetApp recommends that `DBWR_IO_SLAVES` be used for single-CPU systems and that `DB_WRITER_PROCESSES` be used with systems having multiple CPUs.

4.5 FLASHBACK AND FLASH RECOVERY AREA

FLASHBACK

Oracle Flashback maintains a before image record of changed database blocks. Its primary use is to repair user-level errors and incorrect data entry. It acts as a sort of continuous backup. The flashback log can be replayed to restore the database to a point in time. Users might think of it like a “rewind” button for the database. Replaying the flashback log will only restore changed blocks. By default, flashback logs are stored in the flash recovery area.

FLASH RECOVERY AREA

The flash recovery area is an optional storage location that you can use to store recovery-related files such as control file and online redo log copies, archived logs, flashback logs, and RMAN backups. It is an optional Oracle Database-managed directory, file system, or Automatic Storage Management disk group that provides a centralized location for backup and recovery files. You can configure the flash recovery area when creating a database with the Database Configuration Assistant or add it later.

Oracle Database can write archived logs to the flash recovery area. RMAN can store backups in the flash recovery area and restore them from the flash recovery area during media recovery. The flash recovery area also acts as a disk cache for tape.

Oracle Database recovery components interact with the flash recovery area to make sure that the database is completely recoverable by using files stored in the recovery area. All files necessary to recover the database following a media failure are part of the flash recovery area.

The following recovery-related files are stored in the flash recovery area:

- Current control file
- Online redo logs
- Archived redo logs
- Flashback logs
- Control file autobackups
- Data file and control file copies
- Backup pieces

Oracle Database enables you to define a disk limit, which is the amount of space that the database can use in the flash recovery area. It does not include any overhead that is not known to Oracle Database. For example, the disk limit does not include the extra size of a file system that is compressed, mirrored, or uses some other redundancy mechanism.

Oracle Database and RMAN create files in the flash recovery area until the space used reaches the recovery area disk limit. When it needs to make room for new files, Oracle Database deletes files from the flash recovery area that are obsolete, redundant, or backed up to tertiary storage. Oracle Database prints a warning when available disk space is less than 15%, but it continues to fill the disk to 100% of the disk limit.

The bigger the flash recovery area, the more useful it becomes. The recommended disk limit is the sum of the database size, the size of incremental backups, and the size of all archive logs that have not been copied to tape.

Be careful to size the NetApp volume sufficiently large to hold all of the files that will be in the flash recovery area. Regularly monitor space usage in the flash recovery area volume. Setting the Oracle flash recovery disk limit to less than the size of the NetApp volume will help prevent enospc errors and prevent the volume from running out of space.

4.6 ORACLE CLUSTER FILE SYSTEM 2 (OCFS2)

OCFS2 OVERVIEW

OCFS2 is the Oracle open-source file system available on Linux platforms. This is an extent-based (an extent is a variable contiguous space) file system that is currently intended for Oracle data files and Oracle RAC. Unlike the previous release (OCFS), OCFS2 is a general purpose file system that can be used for shared Oracle home installations, making management of Oracle RAC installations even easier. In terms of the file interface it provides to applications, OCFS2 balances performance and manageability by providing functionality that is in between the functionality provided by raw devices and typical file systems. While retaining the performance of raw devices, OCFS2 provides higher order, more manageable file interfaces. In this respect, the OCFS2 service can be thought of as a file system-like interface to raw devices. At the same time, the cluster features of OCFS go well beyond the functionality of a typical file system.

OCFS2 files can be shared across multiple nodes on a network so that the files are simultaneously accessible by all the nodes, which is essential in RAC configurations. For example, sharing data files allows media recovery in case of failures, as all the data files (archive log files) are visible from the nodes that constitute the RAC cluster. Beyond clustering features and basic file service, OCFS2 provides a number of manageability benefits (for example, resizing data files and partitions is easy) and comes with a set of tools to manage OCFS2 files.

OCFS2 CLUSTER CONFIGURATION

The OCFS2 distribution includes of two sets of packages, the kernel module and tools. The kernel module is available for download from http://oss.oracle.com/projects/ocfs2/files and the tools from http://oss.oracle.com/projects/ocfs2-tools/files.

OCFS2 1.2.3 and higher is shipped along with Enterprise Linux 4, while OCFS2 1.2.6 is shipped along with Enterprise Linux 5.

OCFS2 is not distributed by Red Hat. Red Hat Enterprise Linux 4 and 5 users must download and install the appropriate modules and tools packages.

OCFS2 1.2.5 is shipping with SLES10 SP1. OCFS2 1.2.3 is shipping with SLES9 SP3 kernel 2.6.5-7.283 and SLES10.

OCFS2 has a configuration file, `/etc/ocfs2/cluster.conf`. In it, one needs to specify all the nodes participating in the cluster. This file should be the same on all the nodes in the cluster. Whereas one can add new nodes to the cluster dynamically, any other change, for example, name or IP address, requires the cluster to be restarted for the changes to take effect.

OCFS2 tools provide a GUI utility, `ocfs2console,` to set up and propagate the `cluster.conf` to all the nodes in the cluster. This needs to be done only on one of the nodes in the cluster. After this step, users will be able to see the same `/etc/ocfs2/cluster.conf` on all nodes in the cluster.

O2CB CLUSTER SERVICE

OCFS2 comes bundled with its own cluster stack, O2CB. The stack includes:

- NM: Node manager that keeps track of all the nodes in the cluster.conf
- HB: Heartbeat service that issues up/down notifications when nodes join or leave the cluster
- TCP: Handles communication between the nodes
- DLM: Distributed lock manager that keeps track of all locks, its owners and status
- CONFIGFS: User space-driven configuration file system mounted at /config
- DLMFS: User space interface to the kernel space DLM

All the cluster services have been packaged in the o2cb system service. OCFS2 operations, such as format, mount, and so on, require the O2CB cluster service to be at least started in the node where the operation will be performed.

MOUNTING OCFS2 PARTITIONS

Partitioning is recommended even if one is planning to use the entire disk for OCFS2. Apart from the fact that partitioned disks are less likely to be reused by mistake, some features such as mount-by-label only work with partitioned volumes. Use fdisk or parted or any other tool for this task.

Before mounting any disk as OCFS2 partition, it is required to format the disk/LUN using mkfs.ocfs2 or ocfs2console GUI utility.

While formatting, two sizes are required:

- `cluster size`: The sizes supported range from 4K to 1M. For a data file's volume or large files, a cluster size of 128K or larger is appropriate.
- `block size:` The sizes supported range from 512 bytes to 4K. As OCFS2 does not allocate a static node area on format, a 4K block size is recommended for most disk sizes. In contrast, even though it supports 512 bytes, that small a block size is never recommended.

Both the `cluster` and `blocks sizes` are not changeable after the format.

OCFS2 volumes containing the voting disk file (CRS), cluster registry (OCR), data files, redo logs, archive logs, and control files must be mounted with the datavolume and `nointr` mount options. The datavolume option makes sure that Oracle processes open these files with the o_direct flag. The `nointr` option makes sure that the I/Os are not interrupted by signals.

```
# mount -o datavolume,nointr -t ocfs2 /dev/sda1 /u01/db
```

4.7 STORAGE OF ORACLE FILES

Oracle has restrictions on the type of storage that may be used for the Oracle Database 11*g* installation (Oracle Home and CRS Home) and certain Clusterware files. Oracle Clusterware requires two writable files that are shared among RAC nodes. These files are the Oracle Cluster Registry (OCR) file and the voting file.

This section describes the acceptable storage methods for the Oracle software, and the OCR and voting files.

Table 4) Oracle file storage options.

Storage Option	File Types Supported	
	OCR and Voting Disks	Oracle Software
Automatic Storage Management (ASM)	Yes (11*g* R2)	No
OCFS2	Yes	Yes
Red Hat Global File System (GFS); for Red Hat Enterprise Linux and Oracle Enterprise Linux	Yes	Yes
Local storage	No	Yes
Shared disk partitions (block devices)	Yes	No
NFS Client		
Native NFS client	Yes	Yes
Direct NFS client	No	Yes

Note: DNFS is not supported for Oracle Database 11*g* shared CRS files (ORC file, voting file). For databases using Direct NFS client, OCR and voting files should be accessed using another method such as host operating system native NFS.

5 USE CASES

5.1 DATA PROTECTION

BACKUP AND RECOVERY

For additional information about strategies for designing backup, restore, and disaster recovery architectures, see references [8], [9], and [10].

How to Back Up Data from a NetApp System

Data that is stored on a NetApp system can be backed up to online storage, near-line storage, or tape. The protocol used to access data while a backup is occurring must always be considered. When NFS and CIFS are used to access data, Snapshot and SnapMirror can be used and will always result in consistent copies of the file system. They must coordinate with the state of the Oracle Database for database consistency.

With Fibre Channel or iSCSI protocols, Snapshot copies and SnapMirror commands must always be coordinated with the server. The file system on the server must be blocked and all data flushed to the NetApp storage device before invoking the Snapshot command.

Data can be backed up within the same NetApp storage device, to another NetApp storage device, to a NearStore® system, or to a tape storage device. Tape storage devices can be directly attached to an appliance, or they can be attached to an Ethernet or Fibre Channel network, and the appliance can be backed up over the network to the tape device.

Possible methods for backing up data on NetApp systems include:

- Use SnapManager® for Oracle to create your online or offline backups
- Use automated Snapshot copies to create online backups
- Use scripts on the server that can rsh or ssh to the NetApp system to invoke Snapshot copies to create online backups
- Use SnapMirror to mirror data to another NetApp storage device or NearStore system
- Use server operating system–level commands to copy data to create backups
- Use NDMP commands to back up data to a NetApp storage device or NearStore system
- Use NDMP commands to back up data to a tape storage device
- Use third-party backup tools to back up the NetApp storage device or NearStore system to tape or other storage devices

Creating Online Backups Using Snapshot

NetApp Snapshot technology makes extremely efficient use of storage by storing only block-level changes between creating each successive Snapshot copy. Since the Snapshot process is virtually instantaneous, backups are fast and simple. Snapshot copies can be automatically scheduled, they can be called from a script running on a server, or they can be created using SnapDrive® or SnapManager.

Data ONTAP includes a scheduler to automate Snapshot backups. Use automatic Snapshot copies to back up nonapplication data, such as home directories.

Database and other application data should be backed up when the application is in its backup mode. For Oracle Databases this means placing the database tablespaces into hot backup mode prior to creating a Snapshot copy. NetApp has several technical reports that contain details on backing up an Oracle Database.

For additional information on determining data protection requirements, see reference [11].

Best Practice

Use Snapshot copies for performing cold or hot backup of Oracle Databases. No performance penalty is incurred for creating a Snapshot copy. It is recommended to turn off the automatic Snapshot scheduler and coordinate the Snapshot copies with the state of the Oracle Database.

For more information on integrating Snapshot technology with Oracle Database backup, see references [3] and [7].

Recovering Individual Files from a Snapshot Copy

Individual files and directories can be recovered from a Snapshot copy by using native commands on the server, such as the UNIX "`cp`" command, or dragging and dropping in Microsoft® Windows. Data can also be recovered using the single-file SnapRestore command. Use the method that works most quickly.

Recovering Data Using SnapRestore

SnapRestore quickly restores a file system to an earlier state preserved by a Snapshot copy. SnapRestore can be used to recover an entire volume of data or individual files within that volume.

When using SnapRestore to restore a volume of data, the data on that volume should belong to a single application. Otherwise operation of other applications might be adversely affected.

The single-file option of SnapRestore allows individual files to be selected for restore without restoring all of the data on a volume. Be aware that the file being restored using SnapRestore cannot exist anywhere in the active file system. If it does, the appliance will silently turn the single-file SnapRestore into a copy operation. This might result in the single-file SnapRestore taking much longer than expected (normally the command executes in a fraction of a second) and also requires that sufficient free space exist in the active file system.

Best Practice

Use SnapRestore to instantaneously restore an Oracle Database. SnapRestore can restore the entire volume to a point in time in the past or can restore a single file. It is advantageous to use SnapRestore on a volume level, as the entire volume can be restored in minutes, and this reduces downtime while performing Oracle Database recovery. If using SnapRestore on a volume level, it is recommended to store the Oracle log files, archive log files, and copies of control files on a separate volume from the main data file volume and use SnapRestore only on the volume containing the Oracle data files.

For more information on using SnapRestore for Oracle Database restores, see references [3] and [7].

Consolidating Backups with SnapMirror

SnapMirror mirrors data from a single volume or qtree to one or more remote NetApp systems simultaneously. It continually updates the mirrored data to keep it current and available based on the schedule.

SnapMirror is an especially useful tool to deal with shrinking backup windows on primary systems. SnapMirror can be used to continuously mirror data from primary storage systems to dedicated near-line storage systems. Backup operations are transferred to systems where tape backups can run all day long without interrupting the primary storage. Since backup operations are not occurring on production systems, backup windows are no longer a concern.

Creating a Disaster Recovery Site with SnapMirror

SnapMirror continually updates mirrored data to keep it current and available. SnapMirror is the correct tool to use to create disaster recovery sites. Volumes can be mirrored asynchronously or synchronously to systems at a disaster recovery facility. Application servers should be mirrored to this facility as well.

In the event that the DR facility needs to be made operational, applications can be switched over to the servers at the DR site and all application traffic directed to these servers until the primary site is recovered. Once the primary site is online, SnapMirror can be used to transfer the data efficiently back to the production NetApp storage devices. After the production site takes over normal application operation again, SnapMirror transfers to the DR facility can resume without requiring a second baseline transfer.

For more information on using SnapMirror for DR in an Oracle environment, see reference [12].

NDMP and Native Tape Backup and Recovery

The Network Data Management Protocol, or NDMP, is an open standard for centralized control of enterprise-wide data management. The NDMP architecture allows backup application vendors to control native backup and recovery facilities in NetApp appliances and other file servers by providing a common interface between backup applications and file servers.

NDMP separates the control and data flow of a backup or recovery operation into separate conversations. This allows for greater flexibility in configuring the environment used to protect the data on NetApp systems. Since the conversations are separate, they can originate from different locations, as well as be directed to different locations, resulting in extremely flexible NDMP-based topologies. Available NDMP topologies are discussed in detail in reference [13].

If an operator does not specify an existing Snapshot copy when performing a native or NDMP backup operation, Data ONTAP will create one before proceeding. This Snapshot copy will be deleted when the backup completes. When a file system contains FCP data, a Snapshot copy that was created at a point in time when the data was consistent should always be specified. As mentioned earlier, this is ideally done in script by quiescing an application or placing it in hot backup mode before creating the Snapshot copy. After Snapshot copy creation, normal application operation can resume, and tape backup of the Snapshot copy can occur at any convenient time.

When attaching an appliance to a Fibre Channel SAN for tape backup, it is necessary to first make sure that NetApp certifies the hardware and software in use. A complete list of certified configurations is available on the NetApp data protection portal. Redundant links to Fibre Channel switches and tape libraries are not currently supported by NetApp in a Fibre Channel tape SAN. Furthermore, a separate host bus adapter must be used in the NetApp storage device for tape backup. This adapter must be attached to a separate Fibre Channel switch that contains only NetApp storage devices, NearStore appliances, and certified tape libraries and tape drives. The backup server must either communicate with the tape library using NDMP or have library robotic control attached directly to the backup server.

NDMP use cases as applies to NetApp's Data Management Application (DMA) Partners

Some top partners are IBM (TSM) and Symantec (NetBackup™).

Scenario 1 (NDMP Role in Backup of Data from NetApp Storage Device to Tape)

NDMP is used by DMAs to back up data on a NetApp storage device to tape.

NetApp supports several topologies for backup and restore of data. They are local, 3-way, and remote. These topologies differ in the way the tape device is attached to the NetApp storage device.

- **Local:** Tape is attached to the NetApp storage device from where data is to be backed up (NetApp storage device -> tape).
- **3 Way:** Tape is not attached to the NetApp storage device from where data is to be backed up, but to another NetApp storage device on the network (NetApp storage device -> NetApp storage device -> tape).
- **Remote:** Tape is attached to the DMA server, and data from a NetApp storage device is backed up to the tape device on the server (NetApp storage device -> server).

The DMA initiates the backup using NDMP as the protocol and passes on the request to the NDMP data server on the NetApp storage device, which in turn calls DUMP to complete the file enumeration and data transfer. The larger the size and lesser the number of files, the faster is the backup. This is because Data ONTAP updates the DMA with the file history of every file that is being backed up. Enumeration of files before backup and file history updating consume a significant amount of time during the backup process.

Currently, Oracle Secure Backup v10.1.0.3 is certified with Data ONTAP 7.1.1.1 and 7.2.1 on NDMP v4.

Using Tape Devices with NetApp Systems

NetApp storage devices and NearStore systems support backup and recovery from local, Fibre Channel, and Gigabit Ethernet SAN-attached tape devices. Support for most existing tape drives is included as well as a method for tape vendors to dynamically add support for new devices. In addition, the RMT protocol is fully supported, allowing backup and recovery to any capable system. Backup images are written using a derivative of the BSD dump stream format, allowing full file system backups as well as nine levels of differential backups.

Supported Third-Party Backup Tools

NetApp has partnered with a number of third-party vendors to support NDMP-based backup solutions for data stored on NetApp systems. For the complete list, see the Interoperability Matrix located at the NetApp Support site.

Backup and Recovery Best Practices

This section combines the NetApp data protection technologies and products described above into a set of best practices for performing Oracle hot backups (online backups) for backup, recovery, and archival purposes using primary storage (NetApp storage devices with high-performance Fibre Channel disk drives) and near-line storage (NearStore systems with low-cost, high-capacity ATA and SATA disk drives). This combination of primary storage for production databases and near-line disk-based storage for backups of the active data set improves performance and lowers the cost of operations. Periodically moving data from primary to near-line storage increases free space and improves performance, while generating considerable cost savings.

Note: If NetApp NearStore near-line storage is not part of your backup strategy, then refer to [6] for information on Oracle backup and recovery on NetApp storage devices based on Snapshot technology. The remainder of this section assumes both NetApp storage devices and NearStore systems are in use.

SnapManager for Oracle

SnapManager automates and simplifies the complex, manual, and time-consuming processes associated with the backup, recovery, and cloning of Oracle Databases. SnapManager for Oracle leverages NetApp technologies such as Snapshot, SnapRestore, and FlexClone while integrating with the latest Oracle Database releases. SnapManager also integrates with native Oracle technology (such as RAC, RMAN, and ASM) and across FC, iSCSI, and NFS protocols to allow IT organizations to scale their storage infrastructure, meet increasingly stringent SLA commitments, and improve the productivity of database and storage administrators across the enterprise.

With release 3.1, SnapManager for Oracle now also integrates with Oracle Database 11*g* R2 (RAC, RMAN, and ASM) and across FC, iSCSI, NFS, and the new Direct NFS protocols. For the latest supported configurations, see the Interoperability Matrix.

Best Practice
For best practices and requirements of SnapManager for Oracle, see www.netapp.com/us/library/technical-reports/tr-3761.html.

SMO Backup

SnapManager for Oracle utilizes NetApp Snapshot technology to create fast and space-efficient backups. Snapshot copies are point-in-time copies of a database that are created nearly instantaneously. These backups can also be registered with Oracle RMAN, which facilitates the use of RMAN to restore and recover the database at finer granularities such as blocks.

The backups that are created using this technology are space efficient because Snapshot copies consume only enough space to store the differences between the Snapshot copy and the active copy. This also means that DBAs can create more frequent backups, which reduces the mean time to recovery (MTTR).

Another key benefit of SnapManager for Oracle is its ability to verify the integrity of a backup right after performing the backup, using standard Oracle backup verification operations. DBAs can defer the verification of the backup to some later point, if they so desire.

SnapManager for Oracle provides these capabilities for the Oracle Databases in standalone, ASM, and RAC configurations. ASM configurations combine the benefits of storage virtualization of ASM with the efficient backup capabilities of SnapManager. For the RAC configurations, DBAs can invoke SnapManager for Oracle from any RAC database node to create the backup.

SMO Restore and Recover

SnapManager for Oracle restores the database to the state it was in at the time the last Snapshot copy was created. Because the restore process does not involve data movement, the time to restore is a few seconds, regardless of the size of the database. This restore time is significantly less than for traditional recovery methods. Because backups can now be created more frequently, the amount of logs that need to be applied is drastically reduced, thus reducing the MTTR for a database.

SnapManager is also integrated with Oracle RMAN. Thus DBAs can use RMAN to restore and recover the database at finer granularities such as blocks. This integration provides combined benefits of the speed and space efficiency of NetApp Snapshot copies with the fine level of control for restore of Oracle RMAN.

SnapManager for Oracle also provides this capability for ASM-based databases. NetApp has added unique capabilities to Data ONTAP 8 for restoring partial ASM disks to enable this functionality. NetApp has worked with Oracle to develop this solution. It provides fast and efficient restores for the ASM-based databases. In an ASM configuration, an ASM disk group can be shared by multiple databases. As a result, you cannot simply revert to an older Snapshot copy of the disk group, because it would revert all the databases. Traditional restore solutions would go through the host and would require that all the blocks that constitute the database be moved from the storage system to the host and then back to the storage system. The unique solution provided by SnapManager from Oracle relieves this overhead. SnapManager provides the ability to restore just the required data within the ASM disk group without going through the host for most scenarios.

SnapManager provides these restore and recovery capabilities for the Oracle Databases in both standalone and RAC configurations. DBAs can invoke SnapManager for Oracle from any RAC database node to do the restore and recovery. The RAC node where SnapManager performs the restore and recovery operation need not be the same as the node where the backup was performed.

SMO Cloning

A unique feature of SnapManager of Oracle is its ability to automate cloning of Oracle Databases. Using the NetApp FlexClone technology, SnapManager creates writable clones of the Snapshot copy created during backup. Database clones are created quickly, and clones only consume enough storage to hold modified blocks. Because the clone is based on a Snapshot copy, modifying a clone has no impact on the source database. As a result, each developer or QA engineer can be provided with their own personal copy of the database. Developers and QA engineers can make modifications to these personal copies and even destroy them, if needed, without affecting other users.

SnapManager provides these clone capabilities for all configurations of the Oracle Database. This includes both standalone and RAC configurations. Both these configurations can be used along with and without ASM for managing database storage.

5.2 DATA RETENTION: ARCHIVING AND COMPLIANCE (ILM)

CLONING FOR TEST AND DEVELOPMENT ENVIRONMENTS

Development of commercial applications or an in-house development always has a major challenge of how fast the environment can be provided to the developers and QA teams. Some of the challenges faced today are:

- Improve data replication capabilities
- Improve developer productivity
- Meet time-to-market schedules
- Minimize need for new storage
- Automate dev and test environment creation

With the number of developers, test and staging databases keep growing. Making sure that each environment has the right setup at right time is a challenge. Developers also don't have the flexibility to retain point-in-time changes, to enable rollback to previous configuration. If the developer wants to refresh their environment, UNIX copy processes or other procedure need to be initiated, which is time consuming and in turn affects product development schedules.

NetApp introduced the concept of aggregates with flexible volumes (FlexVol volumes) and flexible clones (FlexClone volumes) with Data ONTAP 7G release. This allows thin provisioning and volume resizing on the fly. FlexVol and FlexClone volumes are contained inside an aggregate, which consists of a large number of physical disks grouped into RAID groups using NetApp's double-parity RAID implementation, RAID-DP. RAID-DP is an advanced, cost-effective error protection solution that protects against double disk failure within a single RAID group. FlexClone volumes make use of NetApp Snapshot technology, hence their creation is instantaneous.

Developers can use NetApp FlexClone to make working copies of the preproduction Oracle Database 11*g* database. These cloned copies take up almost no additional storage space. NetApp Snapshot technology allows developers to make point-in-time copies of the data and recover them as needed. The ease of setting up a test and development database systems, especially with FlexClone, is immeasurable compared to traditional UNIX copy or RMAN. FlexClone frees up developers' time, improving their productivity. With traditional procedures, developers have to wait for full restore for hours, in some cases days for the environment to be ready, but with NetApp FlexClone one can clone the environment in minutes as FlexClone never actually copies data so it instantaneous.

Creating Snapshot Copies and FlexClone Volumes

This section describes how to create a Snapshot copy of the volume, create a FlexClone volume, and start the clone Oracle Database 11*g* database on a separate host.

Before creating a Snapshot copy of any Oracle Database 11*g* database volume, first put the database into hot backup mode. Also use the sync command on the host to force any changed blocks to disks.

In our setup we are using three flexible volumes: `oradata`, `oralogs,` and `ora10g`. Since we will only be creating the clone of the database, we would create Snapshot copies of only data files, control files, and online redo log files, which reside in `oradata` and `oralogs` volumes.

Enable `rsh` or `ssh` between the host and the NetApp storage system.

1. To create a Snapshot copy of a volume, following commands can be used:

   ```
   rsh <storage-name> “snap create <volume-name> <snap-name>;”
   rsh Storage “snap create oradata oradata_snap1;”
   rsh Storage “snap create oralogs oralogs_snap1;”
   ```

   You can check the Snapshot copy created using the following commands:

   ```
   rsh Storage “snap list oradata” Volume oradata
   ```

```
working...
%/used %/total date name
---------- ---------- ------------ ------------
0% ( 0%) 0% ( 0%) Jul 16 17:10 oradata_snap1
rsh Storage “snap list oralogs”
Volume oralogs
working...
%/used %/total date name
---------- ---------- ------------ ------------
0% ( 0%) 0% ( 0%) Jul 16 17:12 oralogs_snap1
```

2. After creating Snapshot copies, put the database into normal mode.
3. Before creating a FlexClone volume, check the size of the aggregate on which the parent volume resides:

   ```
   rsh Storage “ df –Ag aggr1;”
   ```

```
Aggregate           total        used         avail        capacity
aggr1               109GB        44GB         65GB         41%
aggr1/.snapshot     5GB          0GB          5GB          0%
```

4. To create a FlexClone volume, use the following commands:

   ```
   rsh <storage-name> “ vol clone create <clone-volume-name> -s none –b <parent- vol-name> <parent-snap-name>
   rsh Storage “ vol clone create oradata_clone –s none –b oradata oradata_snap1;”
   rsh Storage “ vol clone create oralogs_clone –s none –b oralogs oralogs_snap1;”
   ```

During the `vol clone` command, Data ONTAP prints an informational message saying “`Reverting volume vol-name to a previous snapshot.`” For those not familiar with Data ONTAP, this is the standard message when a Snapshot copy is used to restore a volume to a previous state. Since FlexClone volumes leverage Snapshot technology to get a point-in-time image of the parent FlexVol volume, the same mechanism and message are used. The volume mentioned in the message is the new FlexClone volume. Although the word “revert” implies that it is going back to a previous version, it is not actually “reverted,” since it has just come into existence.

5. Check the status of FlexClone volumes using the following command:

   ```
   rsh Storage “vol status oradata_clone;”
   rsh Storage “vol status oralogs_clone;”
   ```

6. Check the space of the aggregate after creating a clone:

   ```
   rsh Storage “ df –Ag aggr1;”
   ```

```
Aggregate           Total        used         avail        capacity
aggr1               109GB        44GB         65GB         41%
aggr1/.snapshot     5GB          0GB          5GB          0%
```

Notice that the amount of space used in the aggregate did not increase. That is because space reservations are by default disabled for FlexClone volumes.

The newly created clone volumes have the same directory structure as that of parent volume.

Make sure that the host is connected to the same NetApp storage and follow all preinstallation OS activities for this node to host the clone Oracle Database 11*g* database.

Mount the FlexClone volumes over NFS. Make sure the local mountpoint name is same as mountpoints used on the primary host that was running Oracle Database 11*g* database.

Copy the `init<sid>.ora` file and create the dump directories in respective folders same as primary RAC database. Since we will be creating a non-RAC clone database from the RAC database, it is required to create a new control file. Also it is required to remove the parameters related to cluster database from the `init<sid>.ora` file.

After starting the clone instance in the nomount stage, create a new control file, recover the database, and then open the database.

ORACLE REAL APPLICATION TESTING WITH NETAPP FLEXCLONE

Oracle Real Application Testing Overview

Before system changes are made, such as hardware and software upgrades, extensive testing is usually performed in a test environment to validate the changes. However, despite the testing, the new system often experiences unexpected behavior when it enters production because the testing was not performed using a realistic workload. The inability to simulate a realistic workload during testing is one of the biggest challenges when validating system changes. Oracle Database 11*g* has a new feature called Database Replay that enables realistic testing of system changes by essentially recreating the production workload environment on a test system. Using Database Replay, one can capture a workload on the production system and replay it on a test system with the exact timing, concurrency, and transaction characteristics of the original workload.

This enables the customer to fully assess the impact of the change, including undesired results, new contention points, or plan regressions. Extensive analysis and reporting are provided to help identify any potential problems, such as new errors encountered and performance divergence.

Capturing the production workload eliminates the need to develop simulation workloads or script, resulting in significant cost reduction and time savings. Also, when Database Replay is used with NetApp FlexClone, realistic testing of complex applications that previously took months using load simulation tools can be completed in days. This enables one to rapidly test changes and adopt new technologies with a higher degree of confidence and at lower risk.

Database Replay performs workload capture of external client workload at the database level and has negligible performance overhead. One can use Database Replay to test any significant system changes, including:

- Database and operating system upgrades
- Configuration changes, such as conversion of a database from a single instance to an Oracle RAC environment
- Storage, network, and interconnect changes
- Operating system and hardware migrations

Oracle Real Application Testing Methods

Database Replay

Following are main steps to be carried out for Database Replay:

- Workload capture
- Workload preprocessing
- Workload replay

Workload Capture

The first step in using database replay is to capture the production workload. Capturing a workload involves recording all requests made by external clients to Oracle Database. When workload capture is enabled, all external client requests directed to Oracle Database are tracked and stored in binary files, called capture files, on the file system. These capture files are platform independent and can be transported to another system. Also, one can specify the start time and duration for the workload capture, as well as the location to store the capture files.

Workload Preprocessing

Once the workload has been captured, the information in the capture files needs to be preprocessed. Preprocessing transforms the captured data into replay files and creates all necessary metadata needed for replaying the workload. This must be done once for every captured workload before they can be replayed. After the captured workload is preprocessed, it can be replayed repeatedly on a replay system running the same version of Oracle Database. Typically, the capture files should be copied to another system for preprocessing. As workload preprocessing can be time consuming and resource intensive, it is recommended that this step be performed on the test system where the workload will be replayed.

Workload Replay

After a captured workload has been preprocessed, it can be replayed on a test system. During the workload replay phase, Oracle Database performs the actions recorded during the workload capture phase on the test system by recreating all captured external client requests with the same timing, concurrency, and transaction dependencies of the production system. Database Replay uses a client program called the replay client to recreate all external client requests recorded during workload capture. Depending on the captured workload, you might need one or more replay clients to properly replay the workload.

Prerequisites for Capturing a Database Workload

Before starting a workload capture, one should have a strategy in place to restore the database on the test system. Before a workload can be replayed, the state of the application data on the replay system should be similar to that of the capture system when replay begins. To accomplish this, NetApp FlexClone is advisable.

1. Put the production database into hot backup mode.
2. Create Snapshot copies of the necessary volumes.
3. Create a FlexClone volume using the Snapshot copies of parent volumes.
4. Mount FlexClone volumes on test system.
5. Recover the database using FlexClone volumes on test system.

This will allow you to restore the database on the replay system to the application state as of the workload capture start time. To create Snapshot copies and FlexClone volumes, see section Creating Snapshot Copies and FlexClone Volume.

Capturing a Database Workload Using APIs

A) Setting Up the Capture Directory

Determine the location and create a directory object in the production database where the captured workload will be stored. Before starting the workload capture, make sure that the directory is empty and has ample disk space to store the workload. If the directory runs out of disk space during a workload capture, the capture will stop. For Oracle RAC, consider using a shared file system. Alternatively, one can set up capture directory paths that resolve to separate physical directories on each instance, but it will be needed to collect the capture files created in each of these directories into a single directory before preprocessing the workload capture.

```
SQL> create or replace directory CAPTURE_DIR as ‘/oradata/capture’;
```

B) Starting Workload Capture

It is important to have a well-defined starting point for the workload so that the replay system can be restored to that point before initiating a replay of the captured workload. To have a well-defined starting point for the workload capture, it is preferable not to have any active user sessions when starting a workload capture. If active sessions perform ongoing transactions, those transactions will not be replayed properly in subsequent database replays, since only that part of the transaction whose calls were executed after the workload capture is started will be replayed.

To start the workload capture, use the `START_CAPTURE` procedure:

```
BEGIN
DBMS_WORKLOAD_CAPTURE.START_CAPTURE (name => ’capture_1’, dir => ’CAPTURE_DIR’,
duration => 600);
END;
/
```

In this example, a workload named `capture_1` will be captured for 600 seconds and stored in the operating system defined by the database directory object named `CAPTURE_DIR`.

The `START_CAPTURE` procedure in this example uses the following parameters:

- The name required parameter specifies the name of the workload that will be captured.
- The dir required parameter specifies a directory object pointing to the directory where the captured workload will be stored.
- The duration optional parameter specifies the number of seconds before the workload capture will end. If a value is not specified, the workload capture will continue until the `FINISH_CAPTURE` procedure is called.

C) Stopping a Workload Capture

To stop the workload capture, use the `FINISH_CAPTURE` procedure:

```
BEGIN
DBMS_WORKLOAD_CAPTURE.FINISH_CAPTURE (); END;
/
```

In this example, the `FINISH_CAPTURE` procedure finalizes the workload capture and returns the database to a normal state.

D) Exporting AWR Data for Workload Capture

Exporting AWR data enables detailed analysis of the workload. This data is also required if one plans to run the AWR compare period report on a pair of workload captures or replays.

To export AWR data, use the `EXPORT_AWR` procedure:

```
BEGIN
DBMS_WORKLOAD_CAPTURE.EXPORT_AWR (capture_id => 2); END;
/
```

In this example, the AWR Snapshot copies that correspond to the workload capture with a capture ID of 2 are exported.

Preprocessing a Database Workload

After a workload is captured and setup of the test system is complete, the captured data must be preprocessed. Preprocessing a captured workload transforms the captured data into replay files and creates all necessary metadata. This must be done once for every captured workload before they can be replayed. After the captured workload is preprocessed, it can be replayed repeatedly on a replay system. To preprocess a captured workload, you will first need to move all captured data files from the directory where they are stored on the capture system to a directory on the instance where the preprocessing will be performed. Preprocessing is resource intensive and should be performed on a system that is:

- Separate from the production system
- Running the same version of Oracle Database as the replay system

For Oracle RAC, select one database instance of the replay system for the preprocessing. This instance must have access to the captured data files that require preprocessing, which can be stored on a local or shared file system. If the capture directory path on the capture system resolves to separate physical directories in each instance, you will need to move all the capture files created in each of these directories into a single directory on which preprocessing will be performed.

To preprocess a captured workload, use the `PROCESS_CAPTURE` procedure:

```
BEGIN
DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE (capture_dir => ’CAPTURE_DIR’); END;
/
```

In this example, the captured workload stored in the dec06 directory will be preprocessed. The PROCESS_CAPTURE procedure in this example uses the `capture_dir r`equired parameter, which specifies the directory that contains the captured workload to be preprocessed.

Replaying a Database Workload

This section describes how to replay a database workload on the test system. After a captured workload is preprocessed, it can be replayed repeatedly on a replay system that is running the same version of Oracle Database.

This section contains the following topics:

- Setting Up the Test System
- Steps for Replaying a Database Workload
- Replaying a Database Workload Using APIs
- Monitoring Workload Replay Using Views

A) Setting Up the Test System

Typically, the replay system where the preprocessed workload will be replayed should be a test system that is separate from the production system. Before a test system can be used for replay, it must be prepared properly as described in the following sections:

- Restoring the database:

  Before a workload can be replayed, the application data state should be logically equivalent to that of the capture system at the start time of workload capture. This minimizes data divergence during replay. The method for restoring the database can be optimally done using NetApp FlexClone as explained in references [2] and [6]. After the database is created with the appropriate application data on the test system, perform the system change you want to test, such as a database or operating system upgrade. The primary purpose of database replay is to test the effect of system changes on a captured workload. Therefore, the system changes you make should define the test you are conducting with the captured workload.
- Resetting the system time:

  It is recommended that the system time on the replay system host be changed to a value that approximately matches the capture start time just before replay is started. Otherwise, an invalid data set might result when replaying time-sensitive workloads.

B) Setting Up Replay Clients

The replay client is a multithreaded program (an executable named wrc located in the `$ORACLE_HOME/bin` directory) where each thread submits a workload from a captured session. Before replay begins, the database will wait for replay clients to connect. At this point, you need to set up and start the replay clients, which will connect to the replay system and send requests based on what has been captured in the workload.

Before starting replay clients, make sure that the:

- Replay client software is installed on the hosts where it will run
- Replay clients have access to the replay directory
- Replay directory contains the preprocessed workload capture
- Replay user has the correct user ID, password, and privileges (the replay user needs the DBA role and cannot be the SYS user)

After these prerequisites are met, you can proceed to set up and start the replay clients using the wrc executable. The wrc executable uses the following syntax:

```
wrc [user/password[@server]] MODE=[value] [keyword=[value]]
```

The parameters user and password specify the username and password used to connect to the host where the wrc executable is installed. The parameter server specifies the server where the wrc executable is installed. The parameter mode specifies the mode in which to run the wrc executable, default is replay.

Starting Replay Clients

After determining the number of replay clients that are needed to replay the workload, one needs to start the replay clients by running the wrc executable in replay mode on the hosts where they are installed. Once started, each replay client will initiate one or more sessions with the database to drive the workload replay. In replay mode, the wrc executable accepts the following keywords:

- userid and password specify the user ID and password of a replay user for the replay client. If unspecified, these values default to the system user.
- server specifies the connection string that is used to connect to the replay system. If unspecified, the value defaults to an empty string.
- replaydir specifies the directory that contains the preprocessed workload capture you want to replay. If unspecified, it defaults to the current directory.

The following example shows how to run the wrc executable in replay mode:

```
%> wrc system/oracle@test mode=replay replaydir=./replay
```

In this example, the wrc executable starts the replay client to replay the workload capture stored in a subdirectory named replay under the current directory. After all replay clients have connected, the database will automatically distribute workload capture streams among all available replay clients. At this point, workload replay can begin. After the replay finishes, all replay clients will disconnect automatically.

C) Replaying a Database Workload Using APIs

This section describes how to replay a database workload using the `DBMS_WORKLOAD_REPLAY` package. Replaying a database workload using the `DBMS_WORKLOAD_REPLAY` package is a multistep process that involves:

- Initializing replay data
- Remapping connections
- Setting workload replay options
- Starting a workload replay

- Stopping a workload replay
- Exporting AWR data for workload replay

Initializing Replay Data

After the workload capture is preprocessed and the test system is properly prepared, the replay data can be initialized. Initializing replay data loads the necessary metadata into tables required by workload replay.

To initialize replay data, use the `INITIALIZE_REPLAY` procedure:

```
BEGIN
DBMS_WORKLOAD_REPLAY.INITIALIZE_REPLAY (replay_name => ’replay_1’, replay_dir =>
’REPLAY_DIR’);
END;
/
```

In this example, the `INITIALIZE_REPLAY` procedure loads preprocessed workload data from the REPLAY_DIR directory into the database. The `INITIALIZE_REPLAY` procedure in this example uses the following parameters:

- The `replay_name` required parameter specifies a replay name that can be used with other APIs to retrieve settings and filters of previous replays.
- The `replay_dir` required parameter specifies the directory that contains the workload capture that needs to be replayed.

Remapping Connections

After the replay data is initialized, connection strings used in the workload capture need to be remapped so that user sessions can connect to the appropriate databases and perform external interactions as captured during replay. To view connection mappings, use the `DBA_WORKLOAD_CONNECTION_MAP` view.

To remap connections, use the REMAP_CONNECTION procedure:

```
BEGIN
DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 101, replay_connection =>
’test:3434/bweb21’);
END;

/
```

In this example, the connection that corresponds to the connection ID 101 will use the new connection string defined by the `replay_connection` parameter. The `REMAP_CONNECTION` procedure in this example uses the following parameters:

- The `connection_id` required parameter is generated when initializing replay data and corresponds to a connection from the workload capture.
- The `replay_connection` optional parameter specifies the new connection string that will be used during workload replay.

Setting Workload Replay Options

After the replay data is initialized and the connections are appropriately remapped, prepare the database for workload replay.

To prepare workload replay on the replay system, use the PREPARE_REPLAY procedure:

```
BEGIN
DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY (synchronization => TRUE); END;
```

```
/
```

In this example, the `PREPARE_REPLAY` procedure prepares a replay that has been previously initialized. The COMMIT order in the workload capture will be preserved. The `PREPARE_REPLAY` procedure uses the following parameters:

- The **`synchronization`** required parameter determines if synchronization will be used during workload replay. If this parameter is set to TRUE, the COMMIT order in the captured workload will be preserved during replay, and all replay actions will be executed only after all dependent COMMIT actions have completed. The default value is TRUE.

Starting a Workload Replay

To start a workload replay, use the START_REPLAY procedure:

```
BEGIN DBMS_WORKLOAD_REPLAY.START_REPLAY (); END;
/
```

Stopping a Workload Replay

To stop a workload replay, use the CANCEL_REPLAY procedure:

```
BEGIN DBMS_WORKLOAD_REPLAY.CANCEL_REPLAY (); END;
/
```

Exporting AWR Data for Workload Replay

Exporting AWR data enables detailed analysis of the workload. This data is also required if you plan to run the AWR compare period report on a pair of workload captures or replays.

To export AWR data, use the EXPORT_AWR procedure:

```
BEGIN
DBMS_WORKLOAD_REPLAY.EXPORT_AWR (replay_id => 1); END;
/
```

In this example, the AWR Snapshot copies that correspond to the workload replay with a replay ID of 1 are exported.

Monitoring Workload Replay Using Views

This section summarizes the views that you can display to monitor workload replay. You need DBA privileges to access these views.

- The DBA_WORKLOAD_CAPTURES view lists all the workload captures that have been captured in the current database.
- The DBA_WORKLOAD_FILTERS view lists all workload filters, for both workload captures and workload replays, defined in the current database.
- The DBA_WORKLOAD_REPLAYS view lists all the workload replays that have been replayed in the current database.
- The DBA_WORKLOAD_REPLAY_DIVERGENCE view enables you to monitor workload replay divergence.

ORACLE DATABASE 11G SECURITY

Stronger Password Protection

This feature modifies the verifier or hash used to store Oracle passwords. This feature provides stronger protection for stored database passwords based on industry standard algorithms and provides enhanced security for password-based authentication by enabling usage of mixed case in passwords. New initialization parameter has been added with Oracle Database 11*g* to secure user accounts.

`SEC_CASE_SENSITIVE_LOGON` controls the case sensitivity in passwords.

TRUE enables case sensitivity.

FALSE disables it.

SYSASM Privilege for ASM

The SYSASM privilege enables the separation of the database operating system credentials from the ASM credentials. Use the SYSASM privilege instead of the SYSDBA privilege to connect to and administer an ASM instance. If you use the SYSDBA privilege to connect to an ASM instance, then Oracle writes warnings to the alert log files because commands that you run using the SYSDBA privilege on an ASM instance will eventually be deprecated. The SYSASM privilege is valid for all platforms. However, operating system authentication varies by platform as follows:

- On UNIX and Linux systems, authentication depends on membership in the operating system group or privilege. In other words, one must be in the DBA group to connect as SYSASM or SYSDBA.
- On Windows systems, Oracle runs as a local system user or administrator. One can connect as SYSASM, SYSDBA, or SYSOPER if you use the local system or administrator credentials.

You can use the following connection types:

- CONNECT ... AS SYSASM to an ASM instance

  Connecting as SYSASM grants you full access to the entire available ASM disk groups.

  ```
  SQLPLUS sys/sys_password AS SYSASM
  ```

- CONNECT ... AS SYSDBA to an ASM instance

  Oracle writes alerts to the alert log files if you issue CREATE, ALTER, or DROP DISKGROUP statements that should be performed by SYSASM.

Tablespace Encryption

Tablespace encryption is an enhancement to the Oracle Advanced Security Transparent Data Encryption solution. Using tablespace encryption, customers can encrypt an entire tablespace, encrypting all data within the tablespace. When the database accesses the tablespace, the relevant data blocks are transparently decrypted for the application. Transparent Data Encryption tablespace encryption provides an alternative to Transparent Data Encryption column encryption by enabling encryption of an entire tablespace. This eliminates the need for granular analysis of applications to determine which columns to encrypt, especially for applications with a large number of columns containing personally identifiable information (PII). Customers who have small amounts of data to encrypt can continue to use the Transparent Data Encryption column encryption solution.

To use tablespace encryption, you should be running Oracle Database 11*g*. If you have upgraded from an earlier release, the compatibility for the database must have been set to 11.0.0 or higher.

Use the following steps to implement tablespace encryption:

1. Set the tablespace master encryption key.
2. Open the Oracle wallet.

3. Create an encrypted tablespace.

Set the Tablespace Master Encryption Key

Setting the tablespace master encryption key is a one-time activity. This creates the master encryption key for tablespace encryption. This key is stored in an external security module (Oracle wallet) and is used to encrypt the tablespace encryption keys.

Tablespace encryption uses the same software wallet that is used by column-based transparent data encryption to store the master encryption key. Check to make sure that the ENCRYPTION_WALLET_LOCATION (or WALLET_LOCATION) parameter in the sqlnet.ora file points to the correct software wallet location. For example, sample sqlnet.ora file would look like following:

```
ENCRYPTION_WALLET_LOCATION = (SOURCE =
                              (METHO = FILE)
                              (METHOD_DATA =
                              (DIRECTORY = /orahome/product/11g/db/wallet)))
```

When one creates a master encryption key for transparent data encryption, a master encryption key for tablespace encryption also gets created.

To set the master encryption key, use the following command:

```
SQL> ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY password
```

Where:

Password is the mandatory wallet password for the security module, with no default setting. It is case sensitive. Enclose the password string in double quotation marks (" ").

When one issues the `ALTER SYSTEM SET ENCRYPTION KEY` command, it recreates the standard transparent data encryption master key if one already exists, and creates a new tablespace master encryption key. If the tablespace master encryption key already exists, a new key is not created.

Open the Oracle Wallet

Before you can create an encrypted tablespace, the Oracle wallet containing the tablespace master encryption key must be open. The wallet must also be open before you can access data in an encrypted tablespace.

```
SQL> STARTUP MOUNT;
SQL> ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY password;
SQL> ALTER DATABASE OPEN;
```

Create an Encrypted Tablespace

The `CREATE TABLESPACE` command enables you to create an encrypted tablespace. It enables one to choose the encryption algorithm and the key length for encryption. The `ENCRYPT` keyword in the `storage_clause` encrypts the tablespace. The following syntax illustrates this:

```
CREATE TABLESPACE securespace

DATAFILE '/home/user/oradata/secure01.dbf' SIZE

150M

ENCRYPTION USING '3DES168'

DEFAULT STORAGE (ENCRYPT);
```

The `ENCRYPTION` keyword is used to specify the encryption algorithm. The ENCRYPT keyword in the `storage_clause` actually encrypts the tablespace. The algorithm can have one of the following values:

`3DES16`

`AES128`

`AES192`

AES256

The key lengths are included in the names of the algorithms themselves. If no encryption algorithm is specified, the default encryption algorithm is used. The default encryption algorithm is AES128.

The following data dictionary views maintain information about the encryption status of a tablespace. One can query these views to verify that a tablespace has been encrypted:

`DBA_TABLESPACES`: The ENCRYPTED column indicates whether a tablespace is encrypted.

`USER_TABLESPACES`: The ENCRYPTED column indicates whether a tablespace is encrypted.

5.3 HIGH AVAILABILITY

FAILOVER CONFIGURATION (FC)

Oracle Data Guard and Snapshot Standby Database

Oracle Data Guard provides high availability, data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production Oracle Databases to survive disasters and data corruptions. Data Guard maintains these standby databases as copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, minimizing the downtime associated with the outage. Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability. There are three types of Data Guard in Oracle Database 11*g*. These are as follows:

Physical Standby Database

Provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. The database schema, including indexes, are the same. A physical standby database is kept synchronized with the primary database, through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database.

As of Oracle Database 11*g* a physical standby database can receive and apply redo while it is open for read-only access. A physical standby database can therefore be used concurrently for data protection and reporting.

Following steps describe how to configure a physical standby database using NetApp storage and NetApp technology.

Step 1: Put the primary database in FORCE LOGGING mode after database creation using the following:

```
SQL statement as sys user:

SQL> ALTER DATABASE FORCE LOGGING;
```

Step 2: Create the init parameter file using server parameter file on the primary database as follows: SQL > create pfile from spfile.

Note: By default it will create the `pfile` on `$ORACLE_HOME/dbs` directory.

Step 3: Add the following contents in to the init parameter file.

```
DB_UNIQUE_NAME=orcl LOG_ARCHIVE_CONFIG='DG_CONFIG=(orcl,orcl1)'
LOG_ARCHIVE_DEST_1='LOCATION=/oradata/arch VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=orcl'
LOG_ARCHIVE_DEST_2='SERVICE=orcl1 ASYNC VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=orcl1' LOG_ARCHIVE_DEST_STATE_1=ENABLE LOG_ARCHIVE_DEST_STATE_2=DEFER
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE LOG_ARCHIVE_FORMAT=%t_%s_%r.arc
LOG_ARCHIVE_MAX_PROCESSES=10
```

Note: orcl and orcl1 is the name of database for primary and standby side, respectively.

Step 4: Create a password file as follows on both primary and standby side.

On Primary Node

```
$ orapwd file=orapworcl  password=oracle ignorecase=Y entries=15
```

On Standby Node

```
$ orapwd file=orapworcl1  password=oracle ignorecase=Y entries=15
```

Step 5: Create listener and tnsnames (having both node entries with service name) file on both primary and standby side.

Note: Make sure tnsping is properly working on both the nodes.

Step 6: Enable archive log on primary database.

Step 7: Shut down primary database and create the Snapshot copy of the volumes (used to store data files, control files, redo log file, and archive log files). Create the FlexClone volume on storage and mount the FlexClone volume on standby side as follows.

Run these commands on the primary node.

```
$ rsh 10.73.69.110 snap create oradata backup

$ rsh 10.73.69.110 vol clone create oradata_clone -b oradata backup –s none

$ rsh 10.73.69.110 exportfs -p rw,anon=0 /vol/oradata_clone

$ rsh 10.73.69.110 exportfs –a
```

Note: 10.73.69.110 is the IP address of NetApp storage and oradata, oradata_clone are the volumes used to store database for primary and standby databases, respectively.

Step 8: Start up primary database in mount mode using init parameter file and take the backup of control file and then open the database using the following command.

```
SQL> startup mount;

SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/orcl1.ctl';

SQL> Alter database open;
```

Step 9: Copy primary database init parameter file to $ORACLE_HOME/dbs of standby side and modify as follows.

```
DB_UNIQUE_NAME=orcl1

LOG_ARCHIVE_CONFIG='DG_CONFIG=(orcl,orcl1)'
LOG_ARCHIVE_DEST_1='LOCATION=/oradata/arch1

VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=orcl1'
LOG_ARCHIVE_DEST_2='SERVICE=orcl ASYNC

VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=orcl'
LOG_ARCHIVE_DEST_STATE_1=ENABLE
```

```
LOG_ARCHIVE_DEST_STATE_2=DEFER REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE

LOG_ARCHIVE_FORMAT=%t_%s_%r.arc

LOG_ARCHIVE_MAX_PROCESSES=10
```

Rename the `Init` parameter file as `initorcl1.ora` instead of `initorcl.ora`.

Step 10: Mount the `oradata_clone` volume with appropriate mount options on standby side. Copy backed-up standby control file from primary node to the standby node.

Rename the copied control file to the proper location (defined in `init` parameter file) of the standby database. Create arch1 directory on oradata directory of standby database. Then use startup mount on standby side.

```
SQL> startup mount;
```

Step 11: Create standby redo log file on stand by database using following command.

```
SQL> alter database add standby logfile '/oradata/orcl/sredo1.log' size 50M;

SQL> alter database add standby logfile '/oradata/orcl/sredo2.log' size 50M;

SQL> alter database add standby logfile '/oradata/orcl/sredo3.log' size 50M;
```

Step 12: Create server parameter file using `init` parameter file on both primary and standby node:

```
SQL> create spfile from pfile;
```

Step 13: On the standby database, issue the following command to start Redo Apply:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
```

Step 14: Verify the physical standby database is performing properly.

1. On the standby database, query the `V$ARCHIVED_LOG` view to identify existing files in the archived redo log.

   For example:

   ```
   Sql > SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME FROM V$ARCHIVED_LOG ORDER BY
   SEQUENCE#
   ```

```
SEQUENCE# FIRST_TIME NEXT_TIME
---------- ------------------ ------------------
8 11-JUL-07 17:50:45 11-JUL-07 17:50:53
9 11-JUL-07 17:50:53 11-JUL-07 17:50:58
10 11-JUL-07 17:50:58 11-JUL-07 17:51:03
3 rows selected.
```

2. Force a log switch to archive the current online redo log file on primary database.

   ```
   SQL> ALTER SYSTEM SWITCH LOGFILE;
   ```

3. Verify the new redo data was archived on the standby database.

   ```
   SQL> SELECT SEQUENCE#, FIRST_TIME, NEXT_TIME FROM V$ARCHIVED_LOG ORDER BY
   SEQUENCE#;
   ```

```
SEQUENCE# FIRST_TIME NEXT_TIME
---------- ------------------ ------------------
8 11-JUL-07 17:50:45 11-JUL-07 17:50:53
9 11-JUL-07 17:50:53 11-JUL-07 17:50:58
10 11-JUL-07 17:50:58 11-JUL-07 17:51:03
11 11-JUL-07 17:51:03 11-JUL-07 18:34:11
4 rows selected.
```

4. Verify new archived redo log files were applied on standby database.

```
SQL > SELECT SEQUENCE#,APPLIED FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
SEQUENCE# APP
```

```
--------- ---
8 YES
9 YES
10 YES
11 YES
4 rows selected.
```

Converting a Physical Standby Database into a Snapshot Standby Database

Stop Redo Apply on the physical standby database as follows.

```
SQL > ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

```
SQL > ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;
```

Logical Standby Database

The logical standby database is kept synchronized with the primary database through SQL

Apply transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database. A logical standby database can be used for other business purposes in addition to disaster recovery requirements. This allows users to access a logical standby database for queries and reporting purposes at any time. Also, using a logical standby database, you can upgrade Oracle Database software and patch sets with almost no downtime. Thus, a logical standby database can be used concurrently for data protection, reporting, and database upgrades.

Following steps describe how to configure a logical standby database using NetApp storage.

1. Prepare the physical standby database as mentioned above.
2. Stop Redo Apply on the physical standby database as follows:

   ```
   SQL > ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
   ```

3. Give the following command on primary database.

   ```
   SQL > alter system set LOG_ARCHIVE_DEST_1='LOCATION=/oradata/arch
   ```

   ```
   VALID_FOR=(ONLINE_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=orcl' scope=both;
   ```

   ```
   SQL > alter system set LOG_ARCHIVE_DEST_3='LOCATION=/oradata/arch2
   ```

   ```
   VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=orcl' scope=both;
   ```

   ```
   SQL > alter system set LOG_ARCHIVE_DEST_STATE_3=ENABLE scope=both;
   ```

4. Build a dictionary in the redo data on primary node as follows.

   ```
   SQL > EXECUTE DBMS_LOGSTDBY.BUILD;
   ```

5. Run the following command to convert to logical standby database on standby node as follows:

   ```
   SQL > ALTER DATABASE RECOVER TO LOGICAL STANDBY orcl1;
   ```

6. Modify logical standby initialization parameters on standby side as follows:

   ```
   SQL > shutdown
   ```

   ```
   SQL> startup mount
   ```

   ```
   SQL > alter system set LOG_ARCHIVE_DEST_1='LOCATION=/oradata/arch1/
   VALID_FOR=(ONLINE_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=orcl1' scope=both;
   ```

```
SQL > alter system set LOG_ARCHIVE_DEST_2='SERVICE=chicago ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=orcl' scope=both;
```

```
SQL > alter system set LOG_ARCHIVE_DEST_3='LOCATION=/oradata/arch3/
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=orcl1'
scope=both;
```

```
SQL > alter system set LOG_ARCHIVE_DEST_STATE_1=ENABLE scope=both; Sql>
alter system set LOG_ARCHIVE_DEST_STATE_2=ENABLE scope=both; Sql> alter
system set LOG_ARCHIVE_DEST_STATE_3=ENABLE scope=both;
```

7. Run the following command to open logical standby database at standby side:

   ```
   SQL> ALTER DATABASE OPEN RESETLOGS;
   ```

8. To apply redo data to the logical standby database, enter the following command. For example:

   ```
   SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
   ```

SNAPMIRROR

There are numerous ways to augment data availability in the countenance of hardware, software, or even site failures. Mirroring provides a mechanism to provide data availability and minimize downtime. SnapMirror performs a block-level mirroring of the data volumes to the destination synchronously or asynchronously, to suit your information availability requirement providing a fast and flexible enterprise solution for mirroring data over local area, wide area and Fibre Channel (FC) networks. SnapMirror can be a key component in implementing enterprise data protection strategies. If a disaster occurs at a source site, business can access mission-critical data from a mirror on the NetApp storage solution at a remote facility, making sure of uninterrupted data availability.

NetApp SnapMirror can be an optimal choice for disaster recovery of Oracle Database systems to provide a back-end store for mission-critical, line of business applications and allow uninterrupted continuity of business and data availability.

Setting Up SnapMirror Relationship

SnapMirror requires a source volume on the primary site and a destination volume on the remote or recovery site for mirroring. The destination volume should be greater than or equal to the size of the source volume.

On the source storage controller use the **snapmirror.access** command to specify the host names of the destination storage system that are allowed to copy data from the source storage system.

Example: **options snapmirror.access host=<destination storage system>**

The destination volumes should be restricted to allow SnapMirror to access them. **vol restrict** command could be used to perform this action. For example: **vol restrict <vol_name>**

Note: **snapmirror status** command could be used to monitor the status of the SnapMirror operations on the specified volumes.

Snapmirror.conf

Snapmirror.conf file is a core configuration of all SnapMirror operations. The /etc/snapmirror.conf file defines the relationship between the source and the destination, schedule intervals used by the destination to copy data, and the arguments that control SnapMirror when copying data. The file resides on the destination NetApp storage systems.

Note: There is a limit of 1,024 entries in the /etc/snapmirror.conf file for each individual storage system. On a storage cluster, the limit applies to the cluster pair. SnapMirror provides three types of mirroring methods:

- Synchronous

- Asynchronous
- Semi-Synchronous

`/etc/snapmirror.conf` file should be used to specify the relationship and the schedule interval which determine the type of replication method. For example:

To configure Synchronous SnapMirror between two volumes:

Edit the **`/etc/snapmirror.conf`** file and type:

```
<src_storage>:<src_vol> <dest_storage>:<dest_vol> - sync
```

The above entry specifies that the **`<Source_Vol>`** on <**`Source_Storage>`** should be replicated to **`<Dest_Vol>`** on **`<Dest_Storage>`** synchronously.

To configure scheduled asynchronous SnapMirror updates between volumes:

Edit the `/etc/snapmirror.conf` file and type:

```
 <src_storage>:<vol_name> <dest_storage>:<vol_name> - <Minute> <Hour> <week of the
month> <day of the week>
```

For example:

```
<src_storage>:<vol_name> <dest_storage>:<vol_name> - 0-59/30 * * *
```

The above example specified that vol_name on src_storage should be replicated to vol_name on dest storage every 30 minutes in an hour every week of the month and every day of the week.

Note: For volumes to be synchronously mirrored using SnapMirror, it is required that the flexible volume is at least 10GB in size.

To configure semi-synchronous SnapMirror between two volumes:

Semi-synchronous mode provides a middle ground that keeps the source and the destination file systems more closely synchronized than in asynchronous mode, but with less application performance impact than synchronous mode. Configuration of the semi-synchronous mode is identical to the configuration of synchronous mode, with the addition of how many writes can be outstanding (unacknowledged by the secondary storage system) before the primary storage system delays acknowledging writes from clients.

Note: If the lag is set to less than 10 seconds, SnapMirror automatically changes to Synchronous mode. It is recommended to set the lag greater that 10 seconds in semi-synchronous mode.

Edit the `/etc/snapmirror.conf` file and type:

```
<src_storage>:<vol_name> <dest_storage>:<vol_name> - semi-sync
outstansding=<seconds> / <milliseconds> / <# of Ops>
```

For example:

```
 <src_storage>:<vol_name> <dest_storage>:<vol_name> - semi-sync
outstansding=10s
```

The above entry specifies that <**`vol_name>`** on <**`src_storage>`** should be replicated to <**`vol_name>`** on

<**`dest_storage>`** semi-synchronously with a lag of 10 seconds.

Understanding the performance impact of SnapMirror on Oracle

Performance in general is a compound and tricky area to quantify. It is certainly beyond the scope of this report to go into overall NetApp storage solution performance, but it makes sense to discuss the effects of SnapMirror on individual storage systems and how it could affect the overall performance.

Any synchronous replication method, regardless of the technology used, will have an impact on the applications using the storage. Hence it becomes important to understand how the business requirements for application performance and data protection will allow an organization to make informed choices between various data protection strategies.

CPU Impact

When a storage system running SnapMirror in synchronous or asynchronous mode receives a write request from a client, the storage system needs to do all the operations that are required normally and do additional processing related to SnapMirror to transfer information to the destination storage, and this adds significant CPU impact on every write operation. Also typically, any read or write activity performed by the clients onto the storage system over the network results in CPU utilization. When replicating data using synchronous or asynchronous mode, all the data written to the primary storage system by clients must be transferred to the destination storage system across the network. So in addition to processing the data from the clients, the storage controller CPU must also process to send the data to the destination storage. So in general one can expect nearly double the CPU utilization on storage systems with synchronous or asynchronous SnapMirror as compared to the same workload on a storage system without SnapMirror.

Network Bandwidth Considerations

All data processed by the source storage system must be replicated to the storage system as it is written; write throughput on the source storage system cannot exceed the network bandwidth available between the source and the destination storage system when using SnapMirror in synchronous or asynchronous mode. It is important to take your network throughput requirements into consideration before sizing network bandwidth requirements between the source and the destination storage systems. On a rough calculation, 250GB of data could be transferred in 7.5 hours over a 10 BaseT full duplex network, and the same might take 1.5 hours on a Gigabit Ethernet link.

Write Latency Impact on Oracle Due to SnapMirror

For each write operation performed on the source storage system, in addition to the normal processing that would occur for the local write without SnapMirror, synchronous and asynchronous modes should do the following:

The source storage system should encapsulate and transmit the operation to the secondary storage system. The secondary storage system should receive it, journal it, and send an acknowledgement to the source storage system.

The source storage system should receive and process the acknowledgement from the secondary storage system.

This process adds latency to each write operation performed by the client on the source storage system.

The amount of latencies can vary based on the operations being performed, the storage system models, and other workload on the source or the destination storage systems.

Write Latency Impact on Oracle Due to Network Distances

In addition to the latencies discussed above, network distances between the source and the destination storage systems could add to it. Over a Fibre Channel network, communications between the source and the destination storage are limited to the speed of light. It is recommended to measure the actual latency on an existing network, rather than make calculations based on theory; typically the speed of light across fibre is estimated to produce 1.05ms of latency per 100 miles of distance. For example, if the source and the destination storage systems are over 100 miles apart, one could expect a round-trip latency of 2.1ms to be added to every write operation performed by the client on the source storage system, while it waits to hear an acknowledgement from the secondary storage system.

Other SnapMirror Factors Adding to Latency on Oracle

Other sources of latency over the network are caused by networking devices the traffic muss pass through, such as routers, switches, and so on. Each device adds latency as it receives the signal on one interface, processes the signal, then transmits through another interface. The amount of latency can be considered small, but can add up if there are many devices in the network.

Choosing the Appropriate Replication Mode

Given the different modes of replication available in SnapMirror and the performance characteristics, it is important to select the correct replication mode for your environment.

Table 5) SnapMirror replication modes.

Replication Mode	Data Lag Time	Bandwidth Requirements	Write Throughput Impact	Write Latency Impact
Asynchronous	Varies on the schedule depending on the configuration	Uses the least bandwidth of all modes	Minimal impact	Minimal impact
Semi-synchronous with outstanding >10 seconds	About 10 seconds in average	Requires bandwidth equal to the write throughput desired on the source storage system	Significant impact on source storage system write throughput, but better than synchronous and semi-synchronous with outstanding <=10 seconds	Minimal impact
Semi-synchronous with outstanding <=10 seconds	Varies from 0 to 10 seconds	Requires bandwidth equal to the write throughput desired on the source storage system, plus a small amount of overhead	Significant impact on the source storage system write throughput	Varies depending on the configuration
Synchronous	Zero lag	Requires bandwidth equal to the write throughput desired on the source storage system, plus a small amount of overhead	Significant impact on the source storage system write throughput	Significant impact on the source storage system write latency

METROCLUSTER

MetroCluster™ is a Fibre Channel site-to-site storage clustering solution to provide high availability and a near-zero RPO disaster recovery back end for applications. MetroCluster uses SyncMirror to maintain a mirror of the primary data between sites without undue latency. MetroCluster is an ideal solution for campus and metropolitan environments.

CFO

Storage controllers can be deployed in active-active pairs for high availability and extra protection. This configuration is generally known as clustered failover (CFO) or HA pair. In a clustered failover pair, each controller services its own workload while monitoring the other controller. If one controller fails, the other controller takes over the workload. The transfer is seamless to users and applications.

The NetApp CFO cluster can be used in conjunction with Oracle RAC to create a high-performance and highly available Oracle Database 11*g* database system.

Load Distribution for Performance

The NetApp CFO cluster consists of a two-node storage cluster. The CFO cluster does not perform automatic load distribution across nodes, so the database files should be distributed across nodes for load balancing. By evenly distributing the database load across storage node, maximum performance can be achieved.

Database Files on Node 1	Database Files on Node 2
Data files (half of the data files, divided by I/O load)	Data files (half of the data files, divided by I/O load)
Temporary files (half of the temporary files, divided by I/O load)	Temporary files (half of the temporary files, divided by I/O load)
Index files (half of the index files, divided by I/O load)	Index files (half of the index files, divided by I/O load)
Redo logs	
	Archived logs

This layout will utilize resources of both storage nodes for maximum performance.

The Oracle home and Oracle CRS home will most likely not have enough I/O to be a significant factor in load distribution across nodes.

The NetApp storage CFO nodes should have following settings:

```
CF.GIVEBACK.AUTO.ENABLE OFF

CF.GIVEBACK.CHECK.PARTNER ON

CF.TAKEOVER.DETECTION.SECONDS 15

CF.TAKEOVER.ON_FAILURE ON

CF.TAKEOVER.ON_NETWORK_INTERFACE_FAILURE ON

CF.TAKEOVER.ON_PANIC ON

CF.TAKEOVER.ON_SHORT_UPTIME ON
```

DATA ONTAP UPGRADE AND NDU

Nondisruptive upgrade (NDU) of Data ONTAP software may be performed on systems running Oracle Database 11*g* using the guidelines in this section.

Upgrade Overview

There are two basic methods of upgrading a Data ONTAP software version. The two methods are disruptive upgrade and nondisruptive upgrade (NDU).

- Disruptive upgrade: This is a Data ONTAP software version upgrade during planned downtime. This involves shutting down all Oracle Databases and applications accessing the storage device, upgrading the Data ONTAP software version, then bringing the databases and applications back online.
- NDU: This is a Data ONTAP software version upgrade while Oracle Databases and applications continue to run and access the storage. This requires an active-active NetApp CFO cluster.

There are two basic types of upgrade. The two types are major version upgrade and minor version upgrade.

- Major upgrade: A major version upgrade is an upgrade from one major release of Data ONTAP to another major release (for example, 7.3.x to 8.0.x).
- Minor upgrade: A minor version upgrade is an upgrade within the same major release family (for example, from 8.0.1 to 8.0.2).

In any case, a disruptive upgrade is much simpler than an NDU. If it's possible to shut down the databases and applications using planned downtime, do so, then upgrade the Data ONTAP software during that downtime. If planned downtime is not an option, use the NDU procedure.

Nondisruptive Upgrade (NDU)

A nondisruptive upgrade (NDU) is to upgrade the Data ONTAP software version running on an active-active clustered pair of NetApp storage device nodes without disrupting I/O to hosts/clients. During NDU, a new software release is downloaded to the boot media on each NetApp storage device node; a negotiated takeover is first performed in one direction; the other node does a clean shutdown and is rebooted; then a giveback and a takeover in the other direction are performed; finally a giveback is performed to get both nodes running the new Data ONTAP version. Each takeover/giveback can take a while, and I/O throughput will be degraded while one NetApp storage device takes the load for two NetApp storage devices. The NDU procedure varies depending on Data ONTAP version, storage protocol, and host multipathing software.

Because the controller for each node in the active-active configuration is connected to both its own disk shelves and the disk shelves of its partner node, a single node can provide access to all volumes or LUNs, even when the partner node is shut down. There are four basic steps in any nondisruptive upgrade:

1. Copy and install the new version of Data ONTAP software to both controllers in the configuration, collect any required configuration information, and make any required changes to the hosts' configurations.
2. Move all I/O to the controller for one node and start the new version of Data ONTAP on the other node.
3. Move all I/O to the controller for the node running the new version and then start the new version of Data ONTAP on the remaining node.
4. Move I/O back to both nodes.

NDU for SAN is more complicated than it is for NAS, and NDU for SAN is further complicated by host multipathing solutions. This is basically because SAN does not readily deal with the idea of a target moving from one physical location to another, that is, no routing. When a NetApp storage device is connected through Fibre Channel to a host, it must be connected using more than one cable and HBA (for HA), and the different paths are resolved into paths to a single LUN by host multipathing software. This is further complicated by the fact that each different host operating system vendor has its own multipathing code.

Preliminary Steps

- Get the latest documentation.
- Get the latest version of the Data ONTAP upgrade guide for the version of Data ONTAP to which you are upgrading. Be aware that NDU might be different with different Data ONTAP versions. You can get this guide from http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml.
- Get the latest Data ONTAP release notes for the version of Data ONTAP to which you are upgrading. The release notes are also available from NetApp Support site.

Check Bugs Online

Search Bugs Online for any known installation or upgrade problems. Be sure to search both the Data ONTAP version to which you are upgrading and the host operating systems of the iSCSI and FCP hosts. Bugs Online is at the NetApp Support site.

Verify the upgrade is supported. Verify the system being upgraded meets the general nondisruptive upgrade requirements documented in the Data ONTAP upgrade guide. This guide lists additional requirements and limitations.

Make sure that your configuration is supported on the appropriate matrix for both the current version of Data ONTAP and the version to which you are upgrading. This includes:

- Host operating system version and patches
- Fibre Channel HBA model and firmware
- Fibre Channel switch model and firmware
- iSCSI initiator version
- Multipath I/O (MPIO) software

See The Compatibility and Configuration Guide for NetApp FCP and iSCSI Products.

It is possible that some of your hosts are supported for nondisruptive upgrade and some are not. You can still complete a nondisruptive upgrade of a system even when some hosts are not supported; those unsupported hosts will be disrupted, of course.

NDU on NAS

For detailed steps to perform NDU on NAS systems, see the Data ONTAP upgrade guide

For information on NDU for NAS systems, see TR-3450: Active/Active Controller Configuration Overview and Best Practice Guidelines.

NDU on SAN

The NDU procedure varies with the Data ONTAP version and with the host software multipathing solution. The latest NDU on SAN support information can be obtained from the NetApp Support site.

NetApp Professional Services also routinely perform NDU on SAN. For contact information for Professional Services, see http://now.netapp.com/NOW/knowledge/docs/san/overviews/ndu.htm.

These references also give detailed information on NDU over SAN:

http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml
www.netapp.com/library/tr/3450.pdf.

5.4 VIRTUALIZATION

VIRTUALIZATION CONCEPTS

Virtualization is an abstraction layer that decouples the physical hardware from the operating system to deliver greater IT resource utilization and flexibility. The adoption of virtualization technology is massive and widespread. Basic virtualization enables multiple virtual machines, each with an operating system and application workload, to run on a single physical server.

Benefits of Virtualization

You can install multiple operating systems on a single physical machine. It can save time installing. Once an OS is installed, other people can copy and use the virtualization image files directly to create their own virtualized OS installations and instances. The really exciting advancement of this technology is found in a virtual infrastructure: the aggregation of servers, storage, and networking components to provide a resource pool of computing and storage capacity.

Disadvantages of Virtualization

Running more operating systems virtually on a single machine can cause the need for more memory and higher performance CPUs. Virtualized environments cannot be migrated to other machines that have different kinds of CPUs, for example, from X86 to ARM. There is currently limited support from Oracle for virtualized environments.

Some virtualization products:

- VMware®
- Hyper-V™
- XEN
- KVM (kernel-based virtual machine for Linux)

XEN

Xen is an open source virtual machine monitor (VMM) for x86-compatible computers. Xen can execute multiple virtual machines, each running its own OS, on a single physical system. Xen is released under the terms of the GNU general public license.

Information on Xen can be obtained from the XenSource Web site: www.xensource.com.

Xen utilizes the paravirtualization method of virtualization. Paravirtualization is a popular virtualization technique that has some similarities to full virtualization. This method uses a hypervisor for shared access to the underlying hardware but integrates virtualization-aware code into the operating system itself. This method does not require any recompilation because the operating systems themselves cooperate in the virtualization process.

Oracle Database 11*g* is not certified for production environments on Xen; however, users might find it useful for test and development environments.

VMWARE

The Oracle Metalink support site has a statement of limited support for Oracle running on VMware environments. Refer to Metalink note 249212.1. In the note Oracle states (in part), "Oracle has not certified any of its products on VMware virtualized environments. Oracle Support will provide support for Oracle products when running on VMware in the following manner: Oracle will only provide support for issues that either are known to occur on the native OS, or can be demonstrated not to be as a result of running on VMware." Refer to the Metalink note for the full text of the statement.

VMware might have limited use in production environments; however, it might be a very popular choice for test and development environments. VMware is a natural fit in NetApp environments because of the fast storage provisioning and cloning made possible by NetApp FlexVol and FlexClone technologies.

Refer to the following NetApp technical reports for more information on using VMware with NetApp storage:

- NetApp and VMware VI3 Storage Best Practices: www.netapp.com/library/tr/3428.pdf
- NetApp and VMware ESX Server 3.0 Building a Virtual Infrastructure from Server to Storage: www.netapp.com/library/tr/3515.pdf

6 APPENDIXES

6.1 OPERATING SYSTEMS

LINUX

Linux: Recommended Versions

The various Linux operating systems are based on the underlying kernel. With all the distributions available, it is important to focus on the kernel to understand features and compatibility.

Kernel Recommendation

The following are the kernel requirements for Oracle Database 11*g* Release 2:

- For Red Hat Enterprise Linux 4.0: Version 2.6.9 or later
- For Oracle Enterprise Linux 5: Version 2.6.18 or later

To determine whether the required kernel is installed, enter the following command:

```
# uname -r
```

The following is a sample output displayed by running this command on Red Hat:

```
Enterprise Linux 4.0 system:

2.6.9-34.EL
```

In this example, the output shows the kernel version (2.6.9) and errata level (34.EL) on the system. If the kernel version does not meet the requirement specified earlier in this section, then contact the operating system vendor for information about obtaining and installing kernel updates.

Linux Version Recommendation

NetApp has tested many kernel distributions, and those based on 2.6.9 are currently recommended. Recommended distributions include Red Hat Enterprise Linux Advanced Server 4.0 and 5.0 as well as Oracle Enterprise Linux 4.0 and 5.0.

Linux: Package Requirements

Oracle Database installation requires the following packages to be installed as a prerequisite. For a complete list of packages, refer to the Software Requirements section of the Oracle 11*g* documentation.

Enterprise Linux 5.0 or Red Hat Enterprise Linux 5.0	The following packages (or later versions) must be installed: binutils-2.17.50.0.6 compat-libstdc++-33-3.2.3 elfutils-libelf-0.125 elfutils-libelf-devel-0.125 elfutils-libelf-devel-static-0.125 gcc-4.1.2 gcc-c++-4.1.2 glibc-2.5-24 glibc-common-2.5 glibc-devel-2.5 glibc-headers-2.5 kernel-headers-2.6.18 ksh-20060214 libaio-0.3.106 libaio-devel-0.3.106 libgcc-4.1.2 libgomp-4.1.2 libstdc++-4.1.2 libstdc++-devel-4.1.2 make-3.81 numactl-devel-0.9.8.i386 sysstat-7.0.2

Linux: OS Settings

The kernel parameter and shell limit values shown in the following section are recommended values only. For production database systems, Oracle recommends that you tune these values to optimize the performance of the system. Refer to the operating system documentation for more information about tuning kernel parameters.

Verify that the kernel parameters shown in the following table are set to values greater than or equal to the recommended value shown. The procedure following the table describes how to verify and set the values.

Table 6) Linux kernel parameters.

Parameter	Value	File
semmsl semmns semopm semmni	250 32000 100 128	/proc/sys/kernel/sem
shmall	2097152	/proc/sys/kernel/shmall

shmmax	Half the size of physical memory (in bytes). For more information, refer to the My Oracle Support Note 567506.1.	/proc/sys/kernel/shmmax
shmmni	4096	/proc/sys/kernel/shmmni
file-max	6815744	/proc/sys/fs/file-max
ip_local_port_range	Minimum: 9000 Maximum: 65500	/proc/sys/net/ipv4/ip_local_port_range
rmem_default	262144	/proc/sys/net/core/rmem_default
rmem_max	4194304	/proc/sys/net/core/rmem_max
wmem_default	262144	/proc/sys/net/core/wmem_default
wmem_max	1048576	/proc/sys/net/core/wmem_max

Note: If the current value for any parameter is higher than the value listed in this table, then do not change the value of that parameter.

To view the current value specified for these kernel parameters and to change them if necessary:

1. Enter the commands shown in the following table to view the current values of the kernel parameters.

Note: Make a note of the current values and identify any values that you must change.

Parameter	Command
semmsl, semmns, semopm, and semmni	# /sbin/sysctl -a | grep sem This command displays the value of the semaphore parameters in the order listed.
shmall, shmmax, and shmmni	# /sbin/sysctl -a | grep shm This command displays the details of the shared memory segment sizes.
file-max	# /sbin/sysctl -a | grep file-max This command displays the maximum number of file handles.
ip_local_port_range	# /sbin/sysctl -a | grep ip_local_port_range This command displays a range of port numbers.
rmem_default	# /sbin/sysctl -a | grep rmem_default
rmem_max	# /sbin/sysctl -a | grep rmem_max
wmem_default	# /sbin/sysctl -a | grep wmem_default
wmem_max	# /sbin/sysctl -a | grep wmem_max

2. If the value of any kernel parameter is different from the recommended value, then complete the following procedure:

   a. Using any text editor, create or edit the /etc/sysctl.conf file, and add or edit lines similar to the following:

   Note: Include lines only for the kernel parameter values that you want to change. For the semaphore parameters (`kernel.sem`), you must specify all four values. However, if any of the current values are larger than the recommended value, then specify the larger value.

```
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
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
```

If you specify the values in the /etc/sysctl.conf file, they persist when you restart the system.

b. Enter the following command to change the current values of the kernel parameters:

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

c. Review the output from this command to verify that the values are correct. If the values are incorrect, edit the `/etc/sysctl.conf` file, then enter this command again.

Setting Shell Limits for the Oracle User

To improve the performance of the software on Linux systems, you must increase the following shell limits for the Oracle user:

Shell Limit	Item in limits.conf	Hard Limit
Maximum number of open file descriptors	nofile	65536
Maximum number of processes available to a single user	nproc	16384

To increase the shell limits:

1. Add the following lines to the `/etc/security/limits.conf` file:

```
2.   oracle       Soft         nproc     2047
3.   oracle       Hard         nproc     16384
4.   oracle       Soft         nofile    1024
5.   oracle       Hard         nofile    65536
6.  oracle        Soft         stack     10240
```

Linux Networking: Full Duplex and Autonegotiation

Most network interface cards use autonegotiation to obtain the fastest settings allowed by the card and the switch port to which it attaches. Sometimes, chipset incompatibilities might result in constant renegotiation or negotiating half duplex or a slow speed. When diagnosing a network problem, be sure the Ethernet settings are as expected before looking for other problems. Avoid hard coding the settings to solve autonegotiation problems, because it only masks a deeper problem. Switch and card vendors should be able to help resolve these problems.

Linux Networking: Gigabit Ethernet Network Adapters

If Linux servers are using high-performance networking (gigabit or faster), provide enough CPU and memory bandwidth to handle the interrupt and data rate. The NFS client software and the gigabit driver reduce the resources available to the application, so make sure resources are adequate. Most gigabit cards that support 64-bit PCI or better should provide good performance.

Any database using NetApp storage should utilize Gigabit Ethernet on both the NetApp storage device and database server to achieve optimal performance.

NetApp has found that the following Gigabit Ethernet cards work well with Linux:

- **SysKonnect.** The SysKonnect SK-98XX series cards work very well with Linux and support single- and dual-fiber and copper interfaces for better performance and availability. A mature driver for this card exists in the 2.4 kernel source distribution.

- **Broadcom**. Many cards and switches use this chipset, including the ubiquitous 3Com solutions. This provides a high probability of compatibility between network switches and Linux clients. The driver software for this chipset appeared in the 2.4.19 Linux kernel and is included in Red Hat distributions with earlier 2.4 kernels. Be sure the chipset firmware is up to date.
- **AceNIC Tigon II**. Several cards, such as the NetGear GA620T, use this chipset, but none are still being manufactured. A mature and actively maintained driver for this chipset exists in the kernel source distribution.
- **Intel® EEPro/1000**. This appears to be the fastest gigabit card available for systems based on Intel, but the card's driver software is included only in recent kernel source distributions (2.4.20 and later) and might be somewhat unstable. The card's driver software for earlier kernels can be found on the Intel Web site. There are reports that the jumbo frame MTU for Intel cards is only 8998 bytes, not the standard 9000 bytes.

Linux Networking: Jumbo Frames with GbE

All of the cards described above support the jumbo frames option of Gigabit Ethernet. Using jumbo frames can improve performance in environments where Linux NFS clients and NetApp systems are together on an unrouted network. Be sure to consult the command reference for each switch to make sure it is capable of handling jumbo frames. There are some known problems in Linux drivers and the networking layer when using the maximum frame size (9000 bytes). If unexpected performance slowdowns occur when using jumbo frames, try reducing the MTU to 8960 bytes.

Linux NFS Protocol: Mount Options

Setting the right NFS mount options can significantly affect both performance and reliability of the I/O subsystem. The NetApp recommended mount options have been thoroughly tested and approved by both NetApp and Oracle.

For the latest NFS mount options and related information, see http://kb.netapp.com/support/index?page=content&id=3010189.

iSCSI Initiators for Linux

iSCSI is simply the pairing of the "best" of both NAS (using NFS or CIFS over IP Ethernet networks) and SAN (Fibre Channel fabrics) technologies. What you ultimately get is a protocol that allows you to use SCSI commands such as Fibre Channel FCP, yet does it over an IP network instead of a costly Fibre Channel fabric. Instead of buying an expensive Brocade or McData switch and costly Fibre Channel HBAs from companies such as JNI, Adaptec, and Emulex, you can use any IP switch (NetGear, 3Com, Extreme, Foundry, and so on) and normal Ethernet cards. Because SCSI is CPU intensive during high I/O loads, iSCSI host bus adapters (HBA) have arrived that act just like an FC HBA except that they use Ethernet instead of FC fabric, the idea being that the SCSI requests are offloaded from your primary CPU onto the ASIC on your iSCSI HBA.

Best Practice
NetApp recommends using the NetApp iSCSI host attach kit for Linux over a high-speed dedicated Gigabit Ethernet network on platforms such as RHEL, OEL, and SUSE with Oracle Databases.

FCP SAN Initiators for Linux

Storage area networks (SANs) have proven to be a promising technology, simplifying the management of large and complex storage systems. But they come at a high cost: First-generation SANs depend on Fibre Channel (FC) networks, which require laying new cables, learning new skills, and buying specialized switches. Unfortunately, this has made SANs hard to justify for all but the largest installations.

Best Practice

NetApp recommends using NetApp FCP Host attach kit 3.0 or later. NetApp recommends using Fibre Channel SAN with Oracle Databases on Linux where there is an existing investment in Fibre Channel infrastructure. NetApp also recommends considering Fibre Channel SAN solutions for Linux when the sustained throughput requirement for the Oracle Database server is more than 1GB per second.

SUN SOLARIS OPERATING SYSTEM

Solaris: Recommended Versions

These are the minimum Solaris versions for Oracle Database 11*g* R2.

Table 7) Solaris recommended versions.

Operating System	Version	Recommended
Solaris	8 or earlier	No
Solaris	9 update 5 or earlier	No
Solaris	9 update 6 or later	No
Solaris	10 update 6 or later	Yes

Best Practice

NetApp recommends the use of Solaris 10 Update 6 and above for Oracle Database 11*g* R2.

Solaris: Kernel Parameters

For kernel parameters and more information on Solaris OS settings, see:

- http://download.oracle.com/docs/cd/E11882_01/install.112/e17163/toc.htm
- www.oracle.com/pls/db112/to_pdf?pathname=install.112/e10848.pdf
- www.oracle.com/pls/db112/portal.portal_db?selected=11&frame=#solaris_installation_guides

Solaris: Packages

The following packages (or later versions) are required for Oracle Database 11*g* R2:

SUNWarc

SUNWbtool

SUNWhea

SUNWlibC

SUNWlibm

SUNWlibms

SUNWsprot

SUNWtoo

SUNWi1of

SUNWi1cs(ISO8859-1)

SUNWi15cs(ISO8859-15)

`SUNWxwfnt`

You might also require additional font packages for Java®, depending on your locale. See http://java.sun.com/j2se/1.4.2/font-requirements.html.

Solaris: Patches

Sun patches are frequently updated, so any list becomes obsolete fairly quickly. The patch levels listed are considered a minimally acceptable level for a particular patch; later revisions will contain the desired fixes but might introduce unexpected issues.

Best Practice
NetApp recommends installing the latest revision of any Sun patch. However, report any problems encountered and back out the patch to the revision specified below to see if the problem is resolved.

These recommendations are in addition to, not a replacement for, the Solaris patch recommendations included in the Oracle installation or release notes.

The following are the list of packages required for Oracle Database 11*g* Release 2.

Installation Type or Product	Requirement
All installations	The following operating system packages (or later versions) must be installed: 120753-06: SunOS 5.10: Microtasking libraries (libmtsk) patch 139574-03: SunOS 5.10 141444-09 141414-02
PL/SQL native compilation, Pro*C/C++, Pro*FORTRAN, Oracle Call Interface, Oracle C++ Call Interface, Oracle XML Developer's Kit (XDK)	Patches for Solaris 10: 119963-14: SunOS 5.10: Shared library patch for C++ 124861-15: SunOS 5.10 Compiler Common patch for Sun C C++ (optional)
Database Smart Flash Cache (an Enterprise Edition only feature)	The following patches are required if you are using the Flash Cache feature: 125555-03 140796-01 140899-01 141016-01 139555-08 141414-10 141736-05

Solaris: Compiler Requirements

The following are the Solaris compiler requirements for Oracle Database 11*g* Release 2.

Installation Type or Product	Requirement
PL/SQL native compilation, Pro*C/C++, Oracle Call Interface, Oracle C++ Call Interface, Oracle XML Developer's Kit (XDK)	Oracle Solaris Studio 12 (C and C++ 5.9)

Solaris Networking: Gigabit Ethernet Network Adapters

Sun provides Gigabit Ethernet cards in both PCI and SBUS configurations. The PCI cards deliver higher performance than the SBUS versions.

NetApp recommends the use of the PCI cards wherever possible.

Any database using NetApp storage should utilize Gigabit Ethernet on both the NetApp storage device and database server to achieve optimal performance.

SysKonnect is a third-party NIC vendor that provides Gigabit Ethernet cards. The PCI versions have proven to deliver high performance.

Make sure that the Sun servers with Gigabit Ethernet interfaces are running with full flow control (some require setting both “send” and “receive” to ON individually).

On a Sun server, set Gigabit flow control by adding the following lines to a startup script (such as one in `/etc/rc2.d/S99`*) or modify these entries if they already exist:

```
ndd -set /dev/ge instance  0

ndd -set /dev/ge ge_adv_pauseRX 1

ndd -set /dev/ge ge_adv_pauseTX 1

ndd -set /dev/ge ge_intr_mode    1

ndd -set /dev/ge ge_put_cfg      0
```

Note: The instance might be other than 0 if there is more than one Gigabit Ethernet interface on the system.

Repeat for each instance that is connected to NetApp storage. For servers using /etc/system, add these lines:

```
set ge:ge_adv_pauseRX=1

set ge:ge_adv_pauseTX=1

set ge:ge_intr_mode=1

set ge_ge_put_cfg=0
```

Note that placing these settings in /etc/system changes every Gigabit interface on the Sun server. Switches and other attached devices should be configured accordingly.

Solaris Networking: Jumbo Frames with GbE

SysKonnect provides SK-98xx cards that do support jumbo frames. To enable jumbo frames, do the following:

1. Edit `/kernel/drv/skge.con`*f* and uncomment this line:

   `JumboFrames_Inst0=”On”;`

2. Edit *`/etc/rcS.d/S50skge`* and add this line:

```
    ifconfig skge0 mtu 9000
```

3. Reboot.

Best Practice
If using jumbo frames with a SysKonnect NIC, use a switch that supports jumbo frames and enable jumbo frame support on the NIC on the NetApp system.

Solaris Networking: Improving Network Performance

Adjusting the following settings can have a beneficial effect on network performance. Most of these settings can be displayed using the Solaris “ndd” command and set by either using “ndd” or editing the /etc/system file.

`/dev/udp udp_recv_hiwat`: Determines the maximum value of the UDP receive buffer. This is the amount of buffer space allocated for UDP received data. The default value is 8192 (8kB). It should be set to 65,535 (64kB).

`/dev/udp udp_xmit_hiwat`: Determines the maximum value of the UDP transmit buffer. This is the amount of buffer space allocated for UDP transmit data. The default value is 8192 (8kB). It should be set to 65,535 (64kB).

`/dev/tcp tcp_recv_hiwat`: Determines the maximum value of the TCP receive buffer. This is the amount of buffer space allocated for TCP receive data. The default value is 8192 (8kB). It should be set to 65,535 (64kB).

`/dev/tcp tcp_xmit_hiwat`: Determines the maximum value of the TCP transmit buffer. This is the amount of buffer space allocated for TCP transmit data. The default value is 8192 (8kB). It should be set to 65,535 (64kB).

`/dev/ge adv_pauseTX 1`: Forces transmit flow control for the Gigabit Ethernet adapter. Transmit flow control provides a means for the transmitter to govern the amount of data sent; "0" is the default for Solaris, unless it becomes enabled as a result of autonegotiation between the NICs.

Best Practice
NetApp strongly recommends that transmit flow control be enabled. Setting this value to 1 helps avoid dropped packets or retransmits, because this setting forces the NIC card to perform flow control. If the NIC gets overwhelmed with data, it will signal the sender to pause. It might sometimes be beneficial to set this parameter to 0 to determine if the sender (the NetApp system) is overwhelming the client.

`/dev/ge adv_pauseRX 1`: Forces receive flow control for the Gigabit Ethernet adapter. Receive flow control provides a means for the receiver to govern the amount of data received. A setting of "1" is the default for Solaris.

`/dev/ge adv_1000fdx_cap 1`: Forces full duplex for the Gigabit Ethernet adapter. Full duplex allows data to be transmitted and received simultaneously. This should be enabled on both the Solaris server and the NetApp system. A duplex mismatch can result in network errors and database failure.

`sq_max_size`: Sets the maximum number of messages allowed for each IP queue (STREAMS synchronized queue). Increasing this value improves network performance. A safe value for this parameter is 25 for each 64MB of physical memory in a Solaris system up to a maximum value of 100. The parameter can be optimized by starting at 25 and incrementing by 10 until network performance reaches a peak.

`Nstrpush`: Determines the maximum number of modules that can be pushed onto a stream and should be set to 9.

Ncsize: Determines the size of the directory name lookup cache (DNLC). The DNLC stores lookup information for files in the NFS-mounted volume. A cache miss might require a disk I/O to read the directory when traversing the pathname components to get to a file.

Cache hit rates can significantly affect NFS performance; getattr, setattr, and lookup usually represent greater than 50% of all NFS calls. If the requested information isn't in the cache, the request will generate a disk operation that results in a performance penalty as significant as that of a read or write request. The only limit to the size of the DNLC is available kernel memory. Each DNLC entry uses about 50 bytes of extra kernel memory.

Best Practice
NetApp recommends that ncsize be set to 8000.

nfs:nfs3_max_threads: The maximum number of threads that the NFS V3 client can use. The recommended value is 24.

nfs:nfs3_nra: The readahead count for the NFS V3 client. The recommended value is 10.

nfs:nfs_max_threads: The maximum number of threads that the NFS V2 client can use. The recommended value is 24.

nfs:nfs_nra: The readahead count for the NFS V2 client. The recommended value is 10.

Solaris IP Multipathing (IPMP)

Solaris IPMP is a multipath terminology used by Sun that provides fault tolerance and load balancing capabilities for network interface cards. This increases the overall bandwidth of the system by balancing load across different interfaces.

The latest information on IPMP support:

- http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/iscsi_support_matrix.shtml
- http://now.netapp.com/matrix/mtx/login.do

Solaris NFS Protocol: Mount Options

Setting the right NFS mount options can significantly affect both performance and reliability of the I/O subsystem. The NetApp recommended mount options have been thoroughly tested and approved by both NetApp and Oracle.

For the latest NFS mount options and related information, see http://kb.netapp.com/support/index?page=content&id=3010189.

Mount options are specified in `/etc/vfstab` for Oracle mounts that occur automatically at boot time. To specify mount options:

1. Edit the `/etc/vfstab`.
2. For each NFS mount participating in a high-speed I/O infrastructure, make sure the mount options specify the recommended mount options.

Note: These values are default NFS settings for Solaris 10 and 9. Specifying them is not actually required but is recommended for clarity.

Hard. The "soft" option should never be used with databases. It might result in incomplete writes to data files and database file connectivity problems. The “hard” option specifies that I/O requests will retry forever in the event that they fail on the first attempt. This forces applications doing I/O over NFS to hang until the required data files are accessible. This is especially important where redundant networks and servers (for example, NetApp clusters) are employed.

Bg. Specifies that the mount should move into the background if the NetApp system is not available to allow the Solaris boot process to complete. **Because the boot process can complete without all the file systems being available, care should be taken to make sure that required file systems are present before starting the Oracle Database processes.**

Intr: This option allows operations waiting on an NFS operation to be interrupted. This is desirable for rare circumstances in which applications utilizing a failed NFS mount need to be stopped so that they can be reconfigured and restarted. If this option is not used and an NFS connection mounted with the "hard" option fails and does not recover, the only way for Solaris to be recovered is to reboot the Sun server.

rsize/wsize: Determines the NFS request size for reads/writes. The values of these parameters should match the values for nfs.udp.xfersize and nfs.tcp.xfersize on the NetApp system. A value of 32,768 (32kB) has been shown to maximize database performance in the environment of NetApp and Solaris. In all circumstances, the NFS read/write size should be the same as or greater than the Oracle block size.

Vers: Sets the NFS version to be used. Version 3 yields optimal database performance with Solaris.

Proto: Tells Solaris to use either TCP or UDP for the connection. Previously UDP gave better performance but was restricted to very reliable connections. TCP has more overhead but handles errors and flow control better. If maximum performance is required and the network connection between the Sun and the NetApp system is short, reliable, and all one speed (no speed matching within the Ethernet switch), UDP can be used. In general, it is safer to use TCP.

Forcedirectio: Forcedirectio was introduced with Solaris 8. It allows the application to bypass the Solaris kernel cache, which is optimal for Oracle. This option should only be used with volumes containing data files. It should never be used to mount volumes containing executables. Using it with a volume containing Oracle executables will prevent all executables stored on that volume from being started. If programs that normally run suddenly won't start and immediately core dump, check to see if they reside on a volume being mounted using "forcedirectio."

Direct I/O is a tremendous benefit. Direct I/O bypasses the Solaris file system cache. When a block of data is read from disk, it is read directly into the Oracle buffer cache and not into the file system cache. Without direct I/O, a block of data is read into the file system cache and then into the Oracle buffer cache, double-buffering the data, wasting memory space and CPU cycles. Oracle does not use the file system cache.

Using system monitoring and memory statistics tools, NetApp has observed that without direct I/O enabled on NFS-mounted file systems, large numbers of file system pages are paged in. This adds system overhead in context switches, and system CPU utilization increases. With direct I/O enabled, file system page-ins and CPU utilization are reduced. Depending on the workload, a significant increase can be observed in overall system performance. In some cases the increase has been more than 20%.

Direct I/O should only be used on mountpoints that house Oracle Database files, not on Oracle executables (ORACLE_HOME, ORA_CRS_HOME) or when doing normal file I/O operations such as "dd." Normal file I/O operations benefit from caching at the file system level.

A single volume can be mounted more than once, so it is possible to have certain operations utilize the advantages of `forcedirectio` while others don't. However, this can create confusion, so care should be taken.

Best Practice

NetApp recommends the use of forcedirectio on selected volumes where the I/O pattern associated with the files under that mountpoint do not lend themselves to NFS client caching. In general these will be data files with access patterns that are mostly random as well as any online redo log files and archive log files. The forcedirectio option should not be used for mountpoints that contain executable files such as the ORACLE_HOME directory. Using the forcedirectio option on mountpoints that contain executable files will prevent the programs from executing properly.

Multiple Mountpoints

To achieve the highest performance, transactional OLTP databases benefit from configuring multiple mountpoints on the database server and distributing the load across these mountpoints.

To accomplish this, create another mountpoint to the same file system on the NetApp storage device. Then either rename the data files in the database (using the ALTER DATABASE RENAME FILE command) or create symbolic links from the old mountpoint to the new mountpoint.

iSCSI Initiators for Solaris

The latest information on iSCSI initiator support:
http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/iscsi_support_matrix.shtml.

Fibre Channel SAN for Solaris

NetApp introduced the industry’s first unified storage appliance capable of serving data in either NAS or SAN configurations. NetApp provides Fibre Channel SAN solutions for all platforms, including Solaris, Windows, Linux, HP/UX, and AIX. The NetApp Fibre Channel SAN solution provides the same manageability framework and feature-rich functionality that have benefited our NAS customers for years.

Customers can choose either NAS or FC SAN for Solaris, depending on the workload and the current environment. For FC SAN configurations, it is highly recommended to use the latest NetApp SAN host attach kit for Solaris. The kit comes with the Fibre Channel HBA, drivers, firmware, utilities, and documentation. For installation and configuration, refer to the documentation that is shipped with the attach kit.

NetApp has validated the FC SAN solution for Solaris in an Oracle environment. Refer to the Oracle integration guide with NetApp FC SAN in a Solaris environment ([6]) for more details. For performing backup and recovery of an Oracle Database in a SAN environment, refer to [7].

The latest information on supported FCP attach kits:
http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/fcp_support.shtml.

AIX OPERATING SYSTEM

To install Oracle Database on IBM-AIX, see:

- IBM AIX with Oracle Database using NetApp Storage in NAS environments
- IBM AIX with Oracle Database using NetApp Storage in SAN environments

HP-UX OPERATING SYSTEM

To install and configure Oracle on HP-UX, see:

- HP-UX with Oracle Database using NetApp Storage in NAS environments
- HP-UX with Oracle Database using NetApp Storage in SAN environments

6.2 ASM

Starting with Oracle Database 10*g* RDBMS provides a new storage mechanism called Automatic Storage Management (ASM), which provides an integrated cluster file system and volume management features. ASM complements the Oracle Database 10*g* RDBMS with both volume and disk management utilities, removing the need for third-party volume management tools while also reducing the complexity of the enterprise architecture. ASM provides simplicity of managing volumes that may be composed of block-based devices (SAN) or file-based devices (NFS). These devices are the underlying storage for the Oracle RDBMS. There are some additional features included in Oracle Database 11*g*.

Please find below the new features introduced for ASM on Oracle Database 11*g*:

1) New SYSASM role for Automatic Storage Management administration

2) Disk group compatibility attributes

3) ASM fast rebalance

4) ASM fast mirror resync

5) New ASMCMD commands and ASMCA for easy storage configuration

6) Automatic Storage Management preferred mirror read

7) ASM variable size extents, scalability, and performance enhancements

8) Automatic Storage Management rolling migration

ASM WITH ISCSI/FCP

ASM provides an integrated file system and volume manager for the database files, built into the Oracle Database kernel. With this capability, ASM provides an alternative to some of the third-party file system and volume management solutions for database storage management tasks, such as creating/laying out databases and managing the use of disk space.

Oracle ASM on NetApp SAN and iSAN storage gives customers an alternative capability for volume management on the Oracle server host using familiar create/alter/drop SQL statements, simplifying the job of DBAs with regard to database storage provisioning.

Load balancing avoids performance bottlenecks by assuring that the I/O workload utilizes all available disk drive resources.

NetApp storage automatically load-balances I/O among the entire disk drives in a WAFL volume. All LUNs and files placed within a single WAFL volume can be assured of utilizing all the volume’s disk drives in a balanced fashion. ASM provides load-balanced I/O across all LUNs or files in an ASM disk group by distributing the contents of each data file evenly across the entire pool of storage in the disk group based on a 1MB stripe size.

When used in combination, NetApp load balancing allows multiple LUNs and file system data to share common disk drives, while reducing the number of LUNs per ASM disk group for improved manageability. ASM further allows load balancing across multiple WAFL volumes or NetApp storage devices. The WAFL file system and RAID 4 are optimized for random small I/O. WAFL makes sure the data is evenly spread across all the disk drives of a RAID group in a volume. This provides optimal read/write performance in high-transaction database environments.

This section discusses guidelines for configuring ASM disks, creating ASM disk groups, and creating databases within ASM, discussing some best practices along the way.

The following recommendations apply regardless of which of the two high-level deployment models above is selected:

- Use host file systems or NFS to store Oracle binaries and other non-ASM-enabled files. In Oracle 10g RAC environments, NFS can be used to simplify deployment using a shared Oracle home.
- Use either NetApp unified storage (Fibre Channel, iSCSI SAN, or NFS/TCPIP NAS) protocols to access the storage for ASM disk groups.
- Combine each NetApp storage device's disk drives into few large FlexVol volumes to minimize administrative overhead.
- Make sure that the LUNs or files within each disk group are balanced in terms of throughput per capacity, so that the I/O throughput available to a disk group from each LUN or file is proportional to the LUN's (or file's) size. In more concrete terms, all LUNs/files within the disk group should supply roughly equivalent I/O operations per second per gigabyte. Generally this will be best accomplished using disk drives with very similar capacity and performance properties throughout the underlying WAFL volume(s).
- Configure several (for example, four) LUNs or files per ASM disk group per FlexVol volume. Configuring multiple LUNs/files can maximize I/O concurrency to the disk group, overcoming any per-LUN limits of the host driver stack (operating system, host bus adapter driver, and/or multipath driver). Even if initially unnecessary, this provisioning can avoid unneeded data movement that would occur if LUNs within the same volume were later added to the disk group.

ASM WITH NFS

Although ASM over SAN storage has been the more popular option, the customer has the choice of deploying ASM over NFS as well. The ASM layer works independently of the NFS client, and therefore the NFS client resiliency and security are not compromised. ASM over NFS could be an option, especially in the SMB segment where the TCO of deploying and managing a SAN might make that not a viable option. The storage administrator can follow a standard method of deploying and managing storage for structured as well as unstructured data. NFS is a ubiquitous protocol, and therefore an Oracle DBA can easily manage the database requirements on NAS storage. NFS does not require personnel with the specialized skills that are required for administering a SAN environment. A customer who has an existing NS infrastructure can easily deploy an Oracle Database 10*g* Real Application Clusters (RAC) environment with ASM without any change in the storage environment.

This is also a feasible option for customers who are comfortable using NFS storage and want to deploy Oracle Database 10*g*/11*g* Standard Edition (SE). This license of the database does not have the tablespace partitioning feature. A good alternative for an NFS environment is spreading tablespaces across multiple volumes or storage using ASM disk groups.

To know more about ASM on NFS, see www.netapp.com/library/tr/3572.pdf.

7 REFERENCES

1. Data ONTAP 7G: The Ideal Platform for Database Applications: www.netapp.com/library/tr/3373.pdf
2. Oracle9*i* for UNIX: Backup and Recovery Using a NetApp Storage Device: www.netapp.com/library/tr/3130.pdf
3. Using the Linux NFS Client with NetApp: Getting the Best from Linux and NetApp: www.netapp.com/library/tr/3183.pdf
4. Installation and Setup Guide 1.0 for Fibre Channel Protocol on Linux: http://now.netapp.com/NOW/knowledge/docs/hba/fcp_linux/fcp_linux10/pdfs/install.pdf
5. Oracle9*i* for UNIX: Integrating with a NetApp Storage Device in a SAN Environment: www.netapp.com/library/tr/3207.pdf
6. Oracle9*i* for UNIX: Backup and Recovery Using a NetApp Storage Device in a SAN Environment: www.netapp.com/library/tr/3210.pdf

7. Data Protection Strategies for NetApp Storage Devices:
 www.netapp.com/library/tr/3066.pdf
8. Data Protection Solutions Overview:
www.netapp.com/library/tr/3131.pdf
9. Simplify Application Availability and Disaster Recovery:
www.netapp.com/partners/docs/oracleworld.pdf
10. Oracle8*i* for UNIX: Providing Disaster Recovery with NetApp SnapMirror Technology:
www.netapp.com/library/tr/3057.pdf
11. ndmpcopy Reference:
http://now.netapp.com/NOW/knowledge/docs/ontap/rel632/html/ontap/dpg/ndmp11.htm#1270498
12. SnapManager for Oracle:
www.netapp.com/us/library/technical-reports/tr-3761.html
13. Best Practices Guide: SnapManager for Oracle:
www.netapp.com/library/tr/3452.pdf
14. Linux (RHEL 4) 64 Bit Performance with NFS, iSCSI, FCP Using an Oracle Database on NetApp Storage:
www.netapp.com/library/tr/3495.pdf
15. Oracle 10*g* Performance: Protocol Comparison on Sun Solaris 10:
www.netapp.com/library/tr/3496.pdf
16. NOW Database Best Practices Index:
http://now.netapp.com/NOW/knowledge/docs/bpg/db/
17. Oracle Database mount options knowledgebase:
http://kb.netapp.com/support/index?page=content&id=3010189
18. Practices Index:
 www.netapp.com/library/tr/3423.pdf
19. Ethernet Storage Best Practices:
www.netapp.com/us/library/technical-reports/tr-3802.html.

8 REVISION HISTORY

Date	Changes	Updated By
October 2007	Original draft	
August 2009	Section 3.6.6 DNFS	Sankar Bose
February 2010	AIX references	Naveen Harsani
April 2010	Mount options	Fredrick Grahn
August 2010	Section 3.2.3 volume size	Naveen Harsani
July 2011	11*g* R2 update	Naveen Harsani

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.

Go further, faster®

NetApp®

www.netapp.com