



Oracle10g™ for UNIX®: Integrating with a Network Appliance™ Filer

Andrew Firebaugh, Network Appliance, Inc.

February 2006 | TR-3353

ARCHIVAL COPY
Contents may be out-of-date

Table of Contents

1. Purpose and Scope	4
2. Assumptions	4
3. Infrastructure	4
3.1 Oracle10g Server Machine.....	5
3.2 Filer.....	5
3.3 Network	5
3.4 Oracle User and Group Accounts	6
3.5 Network	7
3.6 Link Issues.....	8
4. Installing Oracle Home on a Filer	8
4.1 Advantages.....	9
4.2 Issues	9
4.3 Installing Oracle Home onto a Filer from Scratch.....	9
4.4 Moving an Existing Oracle Home onto a Filer.....	10
4.5 Creating an Oracle Home Server on a Filer.....	11
4.6 Supporting Multiple Oracle Versions on a Filer.....	12
5. Running an Oracle10g Instance with Data Files on a Filer	13
5.1 Advantages.....	13
5.2 Considerations.....	13
6. Creating a New Oracle10g Database on a Filer	14
6.1 Creating the Mountpoint	14
6.2 Creating the Initialization File	14
6.3 Starting Up the Oracle Instance	14
7. Moving an Existing Oracle10g Database onto a Filer	15
7.1 Shutting Down the Instance	15
7.2 Creating the Mountpoint	15
7.3 Moving the Oracle Data Files	15
7.4 Starting Up the Instance.....	15
8. Mirroring Certain Oracle Data Files	16

ARCHIVAL COPY
Contents may be out-of-date

9. **Enabling Database Support on the Filer**.....16

10. **Caveats**.....17

ARCHIVAL COPY
Contents may be out-of-date

1. Purpose and Scope

This document describes the steps necessary to integrate single-instance Oracle10g for UNIX with a Network Appliance filer. This paper only covers single-instance databases. Oracle® Real Application Clusters (RAC) will be covered in a separate document. Specifically, we cover the following issues:

- Placing Oracle Home on a filer using NFS
- Creating an Oracle Home server with a filer
- Storing multiple Oracle versions on a filer
- Creating a new database with data files on a filer
- Migrating a database from local disk onto a filer

2. Assumptions

We assume that you are familiar with Oracle10g and the operation of Network Appliance filers. We also assume that you are familiar with the operation of your version of UNIX. All examples in this technical report are from Oracle10g Enterprise Edition, 64-bit version, running under Sun™ Solaris™ V9, 64-bit version. The scripts contained in this paper may require significant modifications to run under your version of UNIX. Although this paper uses Solaris as an example platform, the fundamental principals covered by this paper will apply to database servers running under any version of UNIX.

The sample scripts in this technical report assume the following:

- The name of the filer is "acme."
- The name of the database server is "earth."
- The path to Oracle Home on the database server is "/export/home/oracle/ora10g."
- The path to database files on the database server is "/export/home/data."
- The target volume of Oracle Home on the filer is "acme:/vol/vol1."
- The target volume of database files on the filer is "acme:/vol/vol2."
- The mountpoint of Oracle Home on the database server is "/export/home/oracle/ora10g."
- The mountpoint of the database files on the database server is "/export/home/data."
- The name of the Oracle instance is "ORCL."
- The name of the administrative user account within Oracle is "sysdba."
- The name of the Oracle user account within UNIX is "oracle," and the password of this user is "oracle."
- The name of the Oracle group account within UNIX is "oinstall."

Obviously, you will need to make the appropriate changes to these settings in order to make these scripts work in your environment.

3. Infrastructure

The following items are needed to run Oracle10g for UNIX in conjunction with a filer:

- Oracle10g server machine (running a version of UNIX supported by Oracle and including NFS client software)
- NetApp filer

- Network
- Oracle user and group accounts
- Filer NFS mountpoint
- One or more symbolic links on the Oracle server machine referencing one or more directories or files on the filer (optional, but highly desirable)

3.1 Oracle10g Server Machine

You need Oracle10g running on UNIX. We used Oracle10g Enterprise Edition, 64-bit version running under Sun Solaris 9, 64-bit version. In your installation, be sure that your system satisfies the requirements for running Oracle10g. For more information on this issue, check the Oracle10g installation manual for your target platform.

3.2 Filer

Any NetApp filer running a recent version of Data ONTAP™ will work. At the time of this writing, Data ONTAP 7.0 is the latest version. Any examples are from this version. The NFS license on the filer must be activated, and the NFS protocol must be set up and running.

If you have not already created volume(s) for database files on the filer, it can be done using the vol create command:

```
acme> vol create vol1 14
```

```
acme> vol create vol2 14
```

On the filer, edit the /etc/exports file to include these lines:

```
/vol/vol1 -rw,root=earth,anon=0
```

```
/vol/vol2 -rw,root=earth,anon=0
```

Run this command:

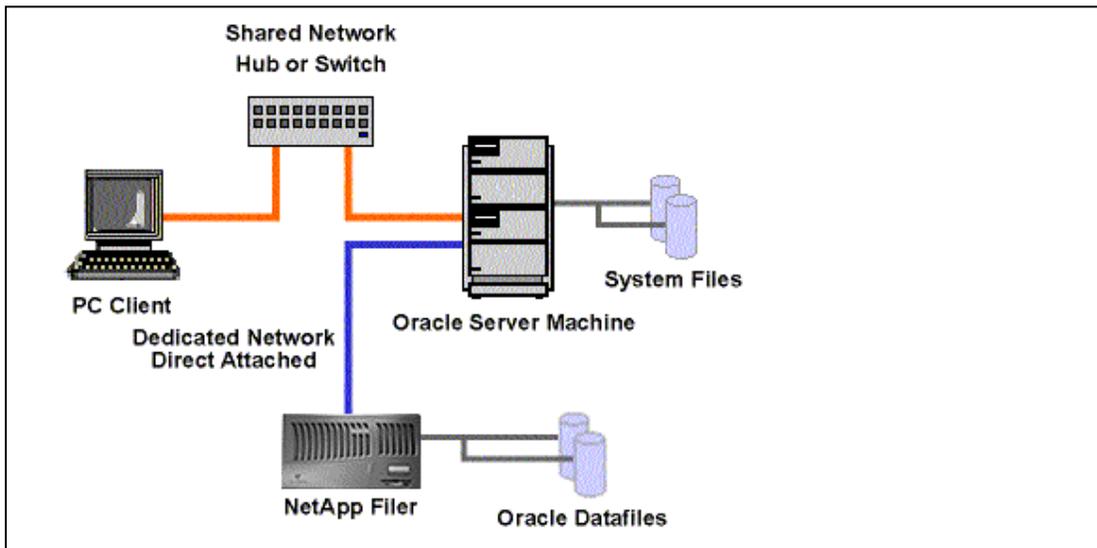
```
acme> exportfs -a
```

This creates volumes with 14 disks and exports them with user access and root privileges from the database server.

3.3 Network

You need a network connection between the Oracle10g server machine and the filer. We have used Fast Ethernet and Gigabit Ethernet, both of which work fine, although Gigabit is preferred. A faster network connection (or possibly multiple network connections) will improve performance, of course.

The following shows the network configuration we used to test this solution:



As shown in the network diagram, we recommend that you dedicate at least one private network connection between the Oracle10g server and the filer. Fast Ethernet and Gigabit Ethernet have both been tested successfully.

A dedicated network connection is not required, but can be used between the Oracle10g server machine and the filer for the following reasons:

- Issues of contention or latency are eliminated or decreased if the Oracle10g server machine and the filer are the only two nodes in the network.
- Security is improved by creating a private network connection. It is easier to protect the Oracle data files from tampering, as would be the case on a shared network.

Each dedicated network connection between the filer and the Oracle10g server machine can be thought of as functionally equivalent to a SCSI or FC-AL connection. (High-end networking protocols such as Gigabit Ethernet provide similar bandwidth as well.)

3.4 Oracle User and Group Accounts

You need to add a user account under which the Oracle processes will run. Normally, this user account is called oracle. This account should be added on both the Oracle10g server machine and the filer. In the Sun Solaris 9 operating system, you can use the "useradd" command to accomplish this task. However, this will affect the Oracle10g server machine only. You need to add an identical user on the NetApp filer. To add the user to the filer, you can copy the appropriate line of the "/etc/passwd" file on the Oracle10g server machine into the "/vol/vol0/etc/passwd" file on the filer. (Or you can use NIS on the filer. For more information on this issue, see the [Data ONTAP System Administrator's Guide](#).)

Add a group account for the Oracle installation. Normally, this group account is called oinstall or dba. We will use oinstall, the default in Oracle10g. This account should be added on both the Oracle10g server machine and the filer. In the Sun Solaris 9 operating system, you can use the "groupadd" command to accomplish this task.

Edit the oracle user's environment file, such as .profile, to include Oracle environment variables such as ORACLE_HOME, ORACLE_BASE, ORACLE_SID, LD_LIBRARY_PATH, and TNS_ADMIN. Also add \$ORACLE_HOME/bin to the oracle users PATH environment variable.

3.5 Network

You need to create a directory on the database server that will be used as a mountpoint for each filer volume being mounted. Make sure the oracle user and group have read, write, and execute permissions on these mountpoints. Security should include:

oracle rwx

For more information on the oracle user account, see Section 3.4: Oracle User and Group Accounts.

Some nondefault NFS mount options should be set when configuring the method that the Oracle10g server machine should use to connect to the filer. The system administrator should define these NFS mounts with the following options (Note: These mount options are for single instance, not RAC):

For Oracle binaries:

-rw,bg,hard,rsize=32768,wsiz=32768,vers=3,nointr,proto=tcp,suid,

For Oracle datafiles and redo logs:

-rw,bg,hard,rsize=32768,wsiz=32768,vers=3,[forcedirectio or lock],nointr,proto=tcp,suid

The latest NetApp NFS mount options for Oracle can be found here:

<http://now.netapp.com/Knowledgebase/solutionarea.asp?id=kb7518>

The following information explains these options in detail.

bg

If the first attempt fails, retry in the background.

hard

This option says that the mountpoint should never time out and that the Oracle10g server machine should not run without it. This will cause the Oracle10g server machine to hang if the filer is not responding to NFS for any reason. If the Oracle10g server machine is booting and the filer is not found, then it will not complete the boot, and Oracle will not start. If it is already up and running, all I/O to and from the filer will suspend until the filer is available again.

intr

This option indicates that the operator wishes to be able to escape out of operations that might hang on this mountpoint. This is sometimes useful.

suid

This option tells the Oracle10g server that it should honor the setuid bit on files mounted at this mountpoint. If you have any of the Oracle10g executables located on the filer, then this setting is important. If you are putting only the database files on the filer, then this option can be omitted. If you use this option, you must also export the file system with the "-anon=0" option. For example, the "/etc/exports" file on the filer should read something like:

```
#Auto-generated by setup Wed Jul 2 14:39:48 PDT 1997
```

```
/vol/vol0 -anon=0,root=earth
```

```
/vol/vol1 -rw,root=earth,anon=0
```

```
/vol/vol2 -rw,root=earth,anon=0
```

See Section 3.2: Filer.

forcedirectio

If **forcedirectio** is specified, then for the duration of the mount, forced direct I/O is used. If the file system is mounted using **forcedirectio**, data is transferred directly between NFS client and server, with no buffering in the NFS client's UNIX kernel page cache. If the file system is mounted using **noforcedirectio**, data is buffered on the client. **forcedirectio** allows all NFS traffic on that mountpoint to bypass the UNIX kernel page cache. Database traffic over NFS is still buffered by the database buffer cache. **forcedirectio** is a performance option that generally benefits applications that do their own buffering, such as the Oracle Database. **forcedirectio** cannot be used on a file system with executable binary files. Therefore, if your Oracle Home is on the filer, you cannot use **forcedirectio** with that file system, but **forcedirectio** can be used with a separate file system that contains data files and log files.

proto

Along with the "vers" parameter, "proto" gives the system administrator the option of choosing whether UDP or TCP should be used. TCP is the supported and recommended protocol. UDP can sometimes lose packets on a noisy network and is therefore not supported for Oracle database files.

3.6 Link Issues

It is often desirable to use symbolic links to map the Oracle server directory structure to the filer. This is true for three reasons:

1. If you are migrating from local disk to a filer, the use of symbolic links prevents you from having to make any modifications to the database.
2. If you later wish to reorganize the location of your Oracle data files, you can do so easily by simply taking the Oracle server down, moving the files, and then editing the link. The symbolic link thus provides an extra layer of abstraction to your file system. Without the symbolic link, you need to edit the database control files, which involves more steps.
3. If you wish to later optimize your Oracle10g installation by adding another dedicated network link between the filer and the Oracle10g server, this can be easily accomplished using a symbolic link. Again, the same thing is possible without the link, but it requires more steps.

One example use of a symbolic link is when Oracle Home is located locally on the database server and a link is created from the local data file path to the data files on the filer. In this example, `$ORACLE_HOME/dbs` links to the location of the data files on the filer. An example of the command used would be:

```
ln -s <filer_mount_point>/data $ORACLE_HOME/dbs
```

4. Installing Oracle Home on a Filer

Oracle Home is the location in the file system where Oracle stores many of its executables and configuration files. On a normal UNIX installation, this is located at something like:

```
/export/home/oracle/ora10g
```

It is sometimes advantageous for Oracle Home to be located on a filer. This configuration creates issues that must be managed as well. The following sections cover these advantages and issues.

4.1 Advantages

Placing Oracle Home on a filer has the following advantages:

- With Oracle Home on a filer, you can easily configure a manual failover database server.
- With Oracle Home on a filer, you can easily set up multiple versions of Oracle on the same server and then easily switch between them. This is ideal in a lab environment, where multiple versions of Oracle must be tested. This is covered in detail in Section 4.6 of this technical report.

4.2 Issues

Placing Oracle Home on a filer creates the following issues:

- You have an additional point of failure in your Oracle installation. If you plan to store Oracle Database files on the filer anyway, this is probably not a serious problem. Loss of the filer would temporarily bring down the database anyway.
- In the event of a filer outage, the error message files that Oracle normally keeps in the "\$ORACLE_HOME/rdbms/logs" directory will not be accessible. Thus, there is a chance that information required to diagnose the problem might not be available. For this reason, you should relocate these files onto the local hard drive of the Oracle10g server machine or onto a separate Network Appliance filer.
- Some Oracle executables require "setuid" permissions. Since these programs need to run with specific permissions, the mountpoint exported by the filer to the Oracle10g server machine must enable this provision. For more information on this issue, see Section 3.5: Filer NFS Mountpoint.

4.3 Installing Oracle Home onto a Filer from Scratch

The procedure for installing Oracle Home from scratch onto a filer requires two simple steps. First, be sure that a reasonable path on the Oracle10g server machine is chosen that resolves to a location on the filer. Second, make sure that this path will automatically mount correctly whenever the Oracle10g server machine is booted.

To ensure this, the system administrator should follow these steps:

To ensure this, the system administrator should follow these steps:

1. Choose a path where Oracle will reside—for example, "/export/home/oracle/ora10g." This becomes the setting for \$ORACLE_HOME.
2. Choose where Oracle Home will reside on the filer—for example, "/vol/vol1" on "acme." Make changes to the Oracle10g server machine so that this point will always be mounted whenever the server boots. Edit the /etc/vfstab file to include the new mount information:

```
acme:/vol/vol1 - /export/home/oracle/ora10g nfs - yes rw,bg,hard,suid,intr
```

3. Mount the file system on the server using the mount command. For Solaris, this would be:

```
# mount /export/home/oracle/ora10g
```

If everything has been defined correctly, the space on the filer will now be available with the correct settings.

4. As the oracle user, create the directory /export/home/oracle/ora10g. Make sure the oracle user has full permissions to this directory.
5. On the Oracle10g server machine, run the Oracle installer as the oracle user and specify the install directory as "/export/home/oracle/ora10g." Since this location is on the filer, the installer will place all of the Oracle executables and configuration files on the filer.

6. Finally, you need to move the alert files and trace files off the filer and onto the Oracle10g server machine's local disks or a different Network Appliance filer. One simple way to do this would be to use a symbolic link as follows:

```
# mkdir /var/oracle
# cd $ORACLE_HOME/rdbms/log
# cp -p * /var/oracle
# cd ..
# rm -rf log
# ln -s /var/oracle log
```

Note: If you are creating an Oracle Home on the filer for the use of multiple database servers, you will need to use another technique to handle the alert and trace file directory. This is covered in Section 4.5 of this technical report.

At this point, the Oracle server software should be installed correctly, with all Oracle data files and executables located on the filer and correctly mounted on the Oracle10g server machine.

4.4 Moving an Existing Oracle Home onto a Filer

Moving an existing Oracle Home installation onto a filer involves the following steps:

1. Shut down the Oracle10g instance.

```
$ sqlplus /nolog
SQL*Plus: Release 10.1.0.2.0 - Production on Thu Oct 21 14:18:53 2004
Copyright (c) 1982, 2004, Oracle. All rights reserved.
```

```
SQL> connect / as sysdba
Connected.
```

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

2. To copy the Oracle files onto the filer, use the following commands:

```
# cd $ORACLE_HOME
# cp -rp * /net/acme/vol/vol1
# cd ..
```

3. To delete the Oracle files on the Oracle10g server machine, enter the following command:

```
# rm -rf $ORACLE_HOME
```

4. Make changes to the Oracle10g server machine so that the NFS file system containing "/vol/vol1" on the filer is mounted to "/export/home/oracle/ora10g" whenever the server boots. Edit the /etc/vfstab file to include the new mount information:

```
acme:/vol/vol1 - /export/home/oracle/ora10g nfs - yes rw,bg,hard,suid,intr
```

5. Mount the file system on the server using the appropriate mount command. For Solaris 9, this would be:

```
# mount /export/home/oracle/ora10g
```

If everything has been defined correctly, the space on the filer is now available with the correct settings. The files that used to be stored at "/export/home/oracle/ora10g" on local disk are now available on the filer at the same path. From the perspective of Oracle, nothing has changed.

6. You need to move the alert files and trace files off the filer and onto the Oracle10g server machine's local disks or a different NetApp filer. One simple way to do this would be:

```
# mkdir /var/oracle
# cd $ORACLE_HOME/rdbms/log
# cp -p * /var/oracle
# cd ..
# rm -rf log
# ln -s /var/oracle log
```

7. Bring the Oracle10g server back up by entering the following command from sqlplus:

```
$ sqlplus /nolog
SQL*Plus: Release 10.1.0.2.0 - Production on Thu Oct 21 14:18:53 2004
Copyright (c) 1982, 2004, Oracle. All rights reserved.
SQL> connect / as sysdba
```

Connected.

```
SQL> startup
```

The Oracle10g instance should come up normally.

4.5 Creating an Oracle Home Server on a Filer

Sharing an ORACLE_HOME over NFS in the single instance context is when multiple single instance database servers share a single ORACLE_HOME on the filer. This is different than a shared ORACLE_HOME in the RAC context, because single instance database servers are not nodes of a cluster and there is no clusterware. Sharing an ORACLE_HOME over NFS in the single instance context is not supported by Oracle for Oracle 10g.

The Oracle document "Oracle Database Installation Guide" (for 10gR1 and 10gR2), Appendix C, "Using NAS Devices," states "for single instance installations (as opposed to RAC installations), you must create a separate Oracle Home directory for each installation. Run the software in this Oracle Home directory only from the system that you used to install it."

Sharing an Oracle Home over NFS for single instance database servers has been a common practice in the past, especially in test and development environments; however, it is not supported with Oracle 10g. There are some advantages to sharing a single instance Oracle Home:

1. Disk space savings.
2. Not necessary to reinstall Oracle for each database server host.
3. Patches and updates need only be applied once.
4. Guaranteed uniformity of the Oracle Home installation used by all hosts.

Disadvantages of sharing an Oracle Home over NFS for single instance database servers:

1. Not supported by Oracle or Network Appliance for Oracle 10g.
2. In order for datafiles of separate database servers to be in separate directories, they must not go in the default datafile directory (ORACLE_HOME/dbs). A separate directory path would need to be used for each database, such as ORACLE_HOME/dbs<sid>.

3. The default listener.ora and tnsnames.ora must be changed with the TNS_ADMIN environment variable, or separate listener names must be used for each host and the default files must be kept up to date with correct entries for all hosts.

Despite the advantages and disadvantages listed above, the practice of sharing an Oracle Home over NFS for single instance database servers is currently not advised for production environments and is currently not supported by Oracle or Network Appliance for Oracle 10g.

A single filer may, however, be used to host multiple single instance Oracle Homes, if there is a separate NFS export (volume or Qtree) used for each Oracle Home. Another recommended method that saves disk space is to install the Oracle Home on a flexible volume, then to create flex clones of the Oracle Home to be used as separate Oracle Homes for other database servers.

To place an Oracle installation on a filer, either you can install Oracle from scratch or you can move an existing set of Oracle executables onto the filer. Both of these techniques are covered in detail in this technical report. Installing Oracle Home from scratch on a filer is covered in Section 4.3, and moving Oracle Home onto a filer is covered in Section 4.4 of this technical report.

In order to avoid a situation where the Oracle diagnostic files are inaccessible due to a failure on the filer, it is advisable to store the Oracle alert and trace files in a local disk directory. To do this, for each Oracle instance that will use Oracle executables on a filer, follow these steps:

1. Shut down the Oracle instance (if it is running).
2. Create a local directory where the alert and trace files will reside (e.g., "/opt/oracle/log").
3. Modify the "BACKGROUND_DUMP_DEST" parameter in the instance's init.ora file to point to the local disk directory. (This parameter controls the target directory for the alert and trace files.)
4. Back up any existing alert and trace files that might exist in the \$ORACLE_HOME/rdbms/log directory on the filer.
5. Start the instance back up again.

At this point, the alert and trace files for this instance will reside on local disk, not on the filer. Thus, a failure on the filer will not make these important diagnostic files inaccessible.

4.6 Supporting Multiple Oracle Versions on a Filer

Similar to Section 4.5 of this technical report, it is possible to store multiple versions of Oracle on a NetApp filer and then bring up Oracle instances that use these different versions of Oracle. For example, you can install both Oracle10g and Oracle9i™ Enterprise Editions on a filer. Create a separate Oracle Home for each version. For example, you can install Oracle10g in "/export/home/oracle/ora10g" and Oracle9i in "/export/home/oracle/ora9i." These paths can be on the same NFS file system or separate file systems. In order to move a database from Oracle9i to Oracle10g, you can follow these steps:

1. Back up your existing database before migrating to a different version.
2. If necessary, export the data in the affected database using the Oracle export utility. When importing data from an Oracle9i database, the Oracle10g database import utility makes appropriate changes to data definitions as it reads earlier releases' export dump files.
3. Shut down the affected instance.
4. Change the environment variables, such as ORACLE_HOME, ORACLE_BASE, ORACLE_SID, LD_LIBRARY_PATH, TNS_ADMIN, and PATH, to point to the new Oracle Home. Verify that \$ORACLE_HOME/bin is part of the oracle users PATH environment variable and verify that it points to the new Oracle Home. It is convenient to make a new environment file, similar to a .profile

- file, such as a "profile.ora10g," for each Oracle Home; that way you can easily source in an entire environment, complete with the appropriate paths, before working with the new installation.
5. Make appropriate changes to the init.ora file to support your new version of Oracle (e.g., the "compatible" parameter).
 6. Bring up the instance in nomount mode.
 7. Run your CREATE DATABASE script. (If you wish to preserve the data files of the other version of Oracle, you should either move them or create the new data files in a different location.)
 8. Import the data using the Oracle import utility or recreate the data by running SQL statements, as appropriate.

Note: Data migration infers moving data from an Oracle version to a more recent Oracle version, for example, from an Oracle9i database to an Oracle10g database. Performing data migration from one version of Oracle to another is a complex process and is beyond the scope of this technical report. For more information on upgrading an Oracle Database and data migration, refer to the Oracle Database Upgrade Guide, available from Oracle.

5. Running an Oracle10g Instance with Data Files on a Filer

There are many advantages to storing Oracle data files on a filer. This configuration creates issues that must be managed as well. The following sections cover these advantages and issues.

5.1 Advantages

Running an Oracle10g instance with data files on a filer has the following advantages:

- Backup performance can be significantly improved. For more information on this issue, refer to TR3130.
- Disk space is abundant on a filer, and, if needed, additional disks can be easily added.
- Administration and tuning requirements are lower when using a Network Appliance filer compared to a local disk configuration. For example, increasing the size of the file system using most local disk setups is a difficult, time-consuming, and complicated process and will usually require significant downtime of the host machine. With a filer, it is a simple operation that takes only a few minutes and requires no downtime to either the filer or the Oracle10g server machine.
- Reliability of the Oracle10g database can be improved. For example, loss of a local disk will typically be more disruptive than loss of a disk on a Network Appliance filer.
- Offloading the I/O to a filer will free up some of the Oracle10g server's CPU capacity.
- Write performance is typically increased by 10% to 30% over well-tuned local disk configurations. (Note: This is very environment specific.) Read performance using a Network Appliance filer should be very close to or slightly better than that of similar local disk configurations.

5.2 Considerations

The following considerations should be kept in mind while running an Oracle instance with the data files residing on a Network Appliance filer:

- In order to achieve point-of-failure recovery, you must ensure that the online redo log files are accessible and up to date in the event of a failure on the filer. For this reason, you should mirror the online redo log files onto local disk (or a second filer if one is available).
- In order to achieve point-of-failure recovery, you must ensure that the control file is accessible and up to date in the event of a failure on the filer. Having the online redo log files alone be accessible

is insufficient if the control file is not available. Thus, you should also mirror the control file onto local disk (or a second filer if one is available).

All of these issues are discussed in Section 8: Mirroring Certain Oracle Data Files. See that section for more information.

6. Creating a New Oracle10g Database on a Filer

This section describes the process of creating a new Oracle10g database with the Oracle10g data files stored on a Network Appliance filer. The steps are the same regardless of whether Oracle Home is located on the Oracle10g server's local disk or on a filer. The database can be created manually using the CREATE DATABASE SQL statement or by using the Database Configuration Assistant (DBCA). Note: The items described in Section 3: Infrastructure are required in order for this to work.

6.1 Creating the Mountpoint

Create a volume on the filer and mount it on the database server for the database files. Make changes to the Oracle10g server machine so that the acme:/vol/vol2 volume on the filer will always be mounted to "/export/home/data" whenever the server boots. For more information, see Section 3.2: Filer, Section 3.5: Filer NFS Mountpoint, and Section 4.3.

Mount the file system on the server using the appropriate mount command. For Solaris, this would be:

```
# mount /export/home/data
```

If everything has been defined correctly, the space on the filer will now be available with the correct settings. Data files and log files can be placed on this NFS file system.

6.2 Creating the Initialization File

Create a database initialization file that places mirrored copies of the control file on both the filer and local disk. The following line in the initialization file accomplishes this:

```
control_files = (/export/home/data/ctl1ORCL.ora, /var/oracle/ctl2ORCL.ora)
```

You can modify the "init.ora" file provided by Oracle for this purpose. If you create a database using DBCA, it will give you the option to mirror control files, at which time you can put one on the filer and one on the local disk

6.3 Starting Up the Oracle Instance

The following steps will manually create a new Oracle10g instance and open it in nomount mode:

1. Set the value of the environment variable ORACLE_SID to the name of the instance you wish to start up.
2. Start Oracle sqlplus.
3. Type the following commands:

```
$ sqlplus /nolog  
connect / as sysdba  
startup nomount  
pfile=/export/home/data/initORCL.ora
```

At that point, the Oracle10g instance should start up. You can now create a database within this instance by running the CREATE DATABASE script. Creating a database in the instance is identical to the steps you would normally take. For more information, see the Oracle10g Installation Guide or the Oracle Database Administrators Guide, available from Oracle.

7. Moving an Existing Oracle10g Database onto a Filer

The following steps will move an existing Oracle10g database from local disk onto a filer. The steps are the same regardless of whether Oracle Home is located on the Oracle server's local disk or on a filer. Note: The items described in Section 3: Infrastructure are required in order for this to work.

7.1 Shutting Down the Instance

Shut down the Oracle10g instance:

```
$ sqlplus /nolog
SQL*Plus: Release 10.1.0.2.0 - Production on Thu Oct 21 14:18:53 2004
Copyright (c) 1982, 2004, Oracle. All rights reserved.

SQL> connect / as sysdba
Connected.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

7.2 Creating the Mountpoint

Make changes to the Oracle10g server machine so that the "/vol/vol2" volume on the filer will always be mounted to "/export/home/data" whenever the server boots. For more information, see Section 3.5: Filer NFS Mountpoint.

Mount the file system on the server using the appropriate mount command. For Solaris, this would be:

```
# mount /export/home/data
```

If everything has been defined correctly, the space on the filer will now be available with the correct settings.

7.3 Moving the Oracle Data Files

Now, move the database files over to the filer. Finally, create a link so that Oracle will be able to find these files when it starts back up. The following sample script accomplishes these tasks:

```
# cd $ORACLE_HOME/dbs
# mv * /export/home/data
# cd ..
# rmdir dbs
# ln -s /export/home/data dbs
```

If you are moving an existing database onto the filer, you need to move all the database files and create links similar to the one shown above. These need not be on a directory basis but may be on a file-by-file basis if your existing files are distributed across many file systems or directories.

7.4 Starting Up the Instance

```
$sqlplus /nolog

SQL*Plus: Release 10.1.0.2.0 - Production on Mon Oct 25 14:27:50 2004

Copyright (c) 1982, 2004, Oracle. All rights reserved.

SQL> connect / as sysdba
Connected to an idle instance.
```

```
SQL> startup open
ORACLE instance started.
```

The database should now be ready for normal operation in its new location.

8. Mirroring Certain Oracle Data Files

If you are running an Oracle10g instance with data files on a filer, you must mirror certain Oracle data files onto local disk if you want to be able to restore from backup up to the point of failure in the event of a failure on the filer. If that is your situation, you should carefully follow the steps in this section.

Shut down your instance and copy the control file to both locations.

Modify the "init.ora" file for your instance to place a mirrored copy of the control file onto local disk. This would involve modifying the following line in the "init.ora" file:

```
control_files = (/export/home/data/ctl1ORCL.ora, /var/oracle/ctl2ORCL.ora)
```

Start up the instance again.

Second, create mirrored sets of online redo log files onto local disk and the filer. For more information on creating mirrored sets of online redo log files, refer to the Oracle Database Administrator's Guide, available from Oracle.

Although the Network Appliance filer is an extremely robust, fault-tolerant device, the approach taken in this technical report is to absolutely ensure the capability to recover the Oracle10g database in the event of a failure on the filer. As long as you keep these issues in mind, you can successfully implement a secure, robust, reliable, and high-performance Oracle10g database with data files on a Network Appliance filer.

9. Enabling Database Support on the Filer

It is a good idea to enable the nvfail feature of Data ONTAP, which supports special error processing when connected to a database server. To do this, enter the following commands from the filer's console or a telnet session:

```
vol options vol1 nvfail on
```

```
vol options vol2 nvfail on
```

This will cause the filer to issue appropriate error messages to the filer's console and in the filer's "/etc/messages" file in the case of an NVRAM failure that might affect the Oracle Database. The administrator can also be notified of these errors by the AutoSupport e-mail notification feature of the filer.

This option also enables some additional status checking when the filer goes through its initialization sequence at boot time to verify that the NVRAM is in a valid state. This should be the case for both a clean (normal) shutdown or a dirty (crash, power failure, etc.) shutdown. Only a failure of the NVRAM card itself should cause it to become invalid. If the content of the NVRAM is found to be invalid, an error message will be put on the system console and into the filer log file, and all attempts by existing NFS clients to access the filer will fail with "stale filehandle" errors. This will affect all NFS clients, including the system running the Oracle server. These stale filehandle errors will cause the Oracle instance to hang or crash, and the Oracle DBA will know that it is necessary to ensure that the state of the database is correct and valid.

Furthermore, additional protection is provided by an optional feature that renames certain files that the system administrator or DBA may wish to ensure are not accessible to the network until after they have been carefully examined. The file "/etc/nvfail_rename" controls this option. If it exists, the files in it are

renamed by having the string ".nvfail" appended to their original filenames in the event of NVRAM failure. Since this occurs before the filer is providing network service, these files will no longer have the same filename as previously. Thus, the applications using them (including an Oracle instance that accesses these files) cannot automatically restart.

The format of the "/etc/nvfail_rename" file is simply the name of the file, one per line, as viewed from the filer, so causing the file:

```
acme:/vol/vol2/data/ctl1ORCL.dbf
```

to be renamed when NVRAM failure is detected. Upon an NVRAM failure being detected, the file will be renamed to:

```
/vol/vol2/home/data/ctl1ORCL.dbf.nvfail
```

This will prevent the Oracle10g server from opening the file and thus make the DBA fully aware of the NVRAM failure.

10. Caveats

For more information regarding use of Network Appliance filers with Oracle, see Using ORACLE with a Multiprotocol Filer by Bruce Clarke. NetApp has not tested with all of the possible combinations of hardware and software options available on UNIX or Solaris. There may be significant differences in your configuration that will alter the procedures necessary to accomplish the objectives outlined in this paper. If you find that any of these procedures does not work in your environment, please contact [the author](#) immediately.

ARCHIVAL COPY
Contents may be out of date