Oracle9i for UNIX:

Integrating with a Network Appliance Filer

Jerry Liu & Sankar Bose | Network Appliance | TR-3129

TECHNICAL REPORT

Network Appliance, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Network Appliance Inc.

Table of Contents

1. Purpose and Scope

This document describes the steps necessary to integrate Oracle9i for UNIX with a Network Appliance filer. 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 instance with datafiles on a filer.
- Migrating an instance from local disk onto a filer.

2. Assumptions

We assume that you are familiar with Oracle9i 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 Oracle9i Enterprise Edition under Sun Solaris Version 8. The scripts contained in this paper may require significant modifications to run under your version of UNIX.

The sample scripts in this technical report assume the following:

- The name of the filer is "acme".
- The source location of Oracle Home on local disk is "/export/home/oracle/".
- The source location of database files on local disk is "${ORACLE_HOME}/dbs".
- The target location of Oracle Home on the filer is "acme:/vol/vol1/home/oracle".
- The target location of database files on the filer is "acme:vol/vol1/home/oracle/dbs".
- The "acme:vol/vol1/home/oracle/dbs" directory's mount point on the Oracle9i Server machine is "/export/home/oracle/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".

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 Oracle9i for UNIX in conjunction with a filer:

- Oracle9i Server machine (running a version of UNIX supported by Oracle and including NFS client software)

- NetApp Filer
- Network
- Oracle user account
- Filer NFS mount point
- 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. Oracle9i Server Machine

You need Oracle9i running on UNIX. We used Oracle9i Enterprise Edition under Sun Solaris Version 8. In your installation, be sure that your system satisfies the requirements for running Oracle9i. For more information on this issue, check the Oracle9i Installation Manual for your target platform.

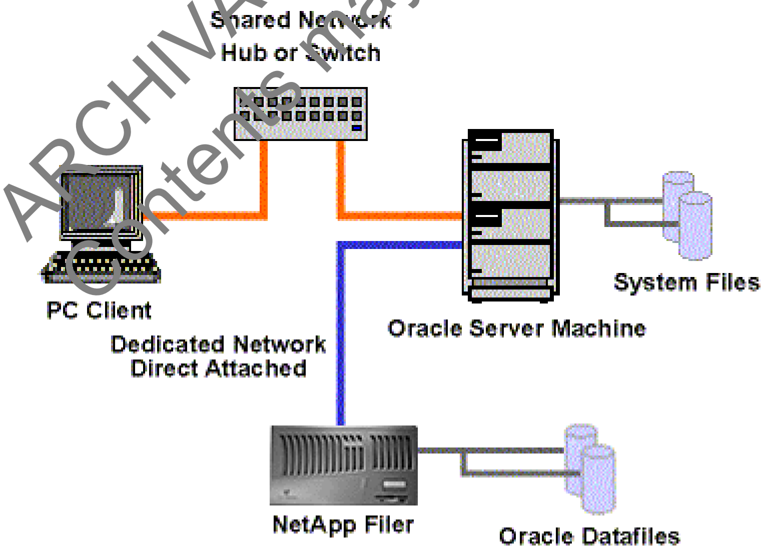
3.2. Filer

Any NetApp filer running Data ONTAP version 5.3 or better will work. The NFS license on the filer must be activated, and the NFS protocol must be setup and running.

3.3. Network

You need a network connection between the Oracle9i server machine and the filer. We have used Fast Ethernet and Gigabit Ethernet, both of which work fine. 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 Oracle9i server and the filer. This can be done using a cross-over cable on most networking topologies (Fast Ethernet and Gigabit Ethernet being two examples we have tested successfully).

Dedicated network connections between the Oracle9i Server machine and the filer are recommended for the following reasons:

- Any issues of contention or latency are eliminated if the Oracle9i Server machine and the filer are the only two nodes in the network.
- Security is ensured by creating a private network connection. There is no issue of protecting the Oracle datafiles from tampering, as would be the case on a shared network.

Each dedicated network connection between the filer and the Oracle9i server machine can be thought of as functionally equivalent to a SCSI or FC-AL connection. (High-end networking protocols like Gigabit Ethernet provide similar bandwidth as well.)

3.4. Oracle User Account

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 Oracle9i Server machine and the filer. In the Sun Solaris 8 operating system, you can use the "useradd" command to accomplish this task. However, this will affect the Oracle9i 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 Oracle9i 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: *Data ONTAP System Administrator's Guide*)

3.5. Filer NFS Mount Point

You need to create a directory on the filer with security set to include:

```
oracle rwx
```

For more information on the oracle user account see 3.4. Oracle User Account.

Some non-default NFS mount options should be set when configuring the method which the Oracle9i Server machine should use to connect to the filer. The System Administrator should define these NFS mounts with the following options:

```
-o hard,intr,suid,vers=3,proto=udp
```

The following table explains these options in detail.

hard This option says that the mount point should never time out and that the Oracle9i Server machine should not run without it. This will cause the Oracle9i Server machine to hang if the filer is not responding to NFS for any reason. If the Oracle9i 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 mount point. This is sometimes useful.

suid This option tells the Oracle9i server that it should honor the set-uid bit on files mounted at this mount point. If you have any of the Oracle9i executables located on the filer, then this setting is important. If you are putting only the database files on the filer, than 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=somepc
/vol/vol1/home -anon=0
```

vers This option is supported in recent releases of UNIX. This option specifies the NFS version which should be used. Some versions of UNIX have been reported to have serious performance problems when running with NFS Version 3. Others perform better using NFS Version 3 instead of Version 2. The System Administrator should try "vers" with both NFS versions, and should run with the NFS version which provides the best performance.

proto Along with the "vers" parameter, "proto" gives the system administrator the option of choosing whether UDP or TCP protocol should be used. For NFS over local area networks, UDP offers slightly less overhead (and therefore better performance) than TCP. However, if your network connection path between the NetApp filer and the Oracle host is prone to lose packets, drop frames, or introduce checksum errors, then TCP can improve performance compared to UDP. (On the other hand, you should not run Oracle with its backend storage being accessed over such an unreliable network.) We therefore recommend that you run using UDP on a dedicated network connection with a cross-over cable between the Oracle9i server and the filer. For more information on this issue, see: 3.3. Network. If you use UDP, be sure to enable UDP checksums on the Oracle9i Server machine.

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 two 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 datafiles, 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 Oracle9i installation by adding another dedicated network link between the filer and the Oracle9i server, this can be easily accomplished using a symbolic link. Again, the same thing is possible without the link, but it requires more steps.

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
```

It is sometimes advantageous for Oracle Home to be located on a filer. This configuration creates issues which 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 fail-over server. This will be the subject of a future technical paper.
- With Oracle Home on a filer, you can run multiple instances of Oracle9i on separate database servers which use the same set of executables. This saves the space otherwise required to store multiple copies of the Oracle9i executables on separate database servers. This is covered in detail in Section 4.5 of this technical report.
- With Oracle Home on a filer, you can easily set up multiple versions of Oracle on the same server, and then 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 Oracle9i Server machine or onto a separate Network Appliance filer.
- The last issue is one of permissions and the use of "setuid" on some of the Oracle executables. Since these programs need to run with specific permissions, the mount point exported by the filer to the Oracle9i Server machine must enable this provision. For more information on this issue, see 3.5. Filer NFS Mount Point

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 Oracle9i Server machine is chosen which resolves to a location on the filer. Second, make sure that this path will automatically mount correctly whenever the Oracle9i Server machine is booted.

To ensure this, the System Administrator should follow these steps:

1. Choose a path where Oracle will reside. For example, "/export/home/oracle". This becomes the setting for "${ORACLE_HOME}".
2. Choose where Oracle Home will reside on the filer. For example, "/vol/vol1/home/oracle" on "acme". On the Oracle9i server machine, this is "acme:/vol/vol1/home/oracle". Make changes to the Oracle9i Server machine so that this point will always be mounted whenever the server boots. For more information on this issue, see 3.5. Filer NFS Mount Point.
3. Mount the file system on the server using the *mount* command. For Solaris, this would be:

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

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

4. On the Oracle9i Server machine run the Oracle Installer and specify the install directory as "/export/home/oracle". Since this is the location of the filer's file system, the installer will place all of the Oracle executables and configuration files on the filer.

5. Finally, you need to move the alert files and trace files off the filer and onto the Oracle9i Server machine's local disks or a different Network Appliance 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
```

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 datafiles and executables located on the filer and correctly mounted on the Oracle9i 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. Shutdown the Oracle9i instance:

```
$ sqlplus

SQL*Plus: Release 9.0.1.0.0 - Production on Mon Sep 24 13:48:43 2001

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Enter user-name: connect / as sysdba
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production

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 -p * /net/acme/vol/vol1/home/oracle
# cd ..
```

3. To delete the Oracle files on the Oracle9i Server machine, enter the following command:

```
# rm -rf ${ORACLE_HOME}
```

4. Make changes to the Oracle9i Server machine so that the NFS file system containing "/vol/vol1/home/oracle" on the filer is mounted to "/export/home/oracle" whenever the server boots. For more information see 3.5. Filer NFS Mount Point.

5. Mount the file system on the server using the appropriate mount command. For Solaris 8, this would be:

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

   If everything has been defined correctly, the space on the filer is now available with the correct settings. The files which used to be stored at "/export/home/oracle" on local disk are now available on the filer in the same location in the directory tree. 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 Oracle9i 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 Oracle9i Server back up by entering the following command from sqlplus connect as sysdba:

   ```
   startup
   ```

   The Oracle9i instance should come up normally.

4.5. Creating an Oracle Home Server on a Filer

The distribution of Oracle9i Enterprise Edition requires more than 2 GB of disk space for a full installation. In a database testing lab, a customer may require tens or even hundreds of Oracle database servers to be running. Each of these servers needs almost a gigabyte of space to store the Oracle executables. This constitutes a waste of disk space, potentially tens or even hundreds of gigabytes.

A NetApp filer can be used to store a common set of Oracle executables which are used by multiple Oracle database servers. In the process, you can save the disk space required to support multiple separate installations of Oracle.

To place an Oracle installation on a filer, you can either 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 which 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 which 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 which use these versions of Oracle. For example, the author has installed both Oracle9i and Oracle8i Enterprise Edition on a filer. In order to flip a server between versions of Oracle you can follow these steps:

1. If necessary, export the data in the affected instance using the Oracle Export utility. (Be aware of incompatibilities in the ROWID type columns when exporting from Oracle7 to Oracle8i and Oracle9i or vice versa.)

2. Shutdown the affected instance.

3. Change the $ORACLE_HOME and $ORACLE_BASE environment variables to point to the new Oracle Home.

4. Make appropriate changes to the init.ora file to support your new version of Oracle (e.g., the "compatible" parameter.)

5. Bring up the instance in nomount mode.

6. Run your CREATE DATABASE script. (If you wish to preserve the datafiles of the other version of Oracle, you should either back them up, or create new datafiles in a different location.)

7. 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 Oracle7 database to Oracle8, Oracle8i, and Oracle9i database. Performing a migration from one version of Oracle to another is a complex process, and is beyond the scope of this technical report. However, it is also possible to perform a migration of a database from one version to another using a set of Oracle executables which are stored on a Network Appliance filer.

5. Running an Oracle9i Instance with Datafiles on a Filer

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

5.1. Advantages

Running an Oracle9i instance with datafiles on a filer has the following advantages:

- Backup performance can be significantly improved. For more information on this issue, refer to TR3130
- 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 and complicated process, usually involving a reboot of the host machine. With a filer, it is a simple operation which takes only a few minutes, and requires no downtime to either the filer or the Oracle9i server machine.
- Reliability of the Oracle9i 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 network interface will free up some of the Oracle9i server's CPU capacity.
- Write performance is typically increased by 10% - 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, similar local disk configurations.

5.2. Considerations

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

- In order to achieve point-of-failure recovery, you must ensure that the on-line 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 on-line 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 on-line 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 Datafiles. See that section for more information.

6. Creating a New Oracle9i Instance on a Filer

This section describes the process of creating a new Oracle9i instance with the Oracle9i data files stored on a Network Appliance filer. The steps are the same regardless of whether Oracle Home is located on the Oracle9i server's local disk or on a filer. **Note:** The items described in the section 3 (viz. Infrastructure section) are required in order for this to work.

6.1. Creating the Mount Point

Make changes to the Oracle9i Server machine so that the "acme:/vol/vol1/home/oracle/dbs" directory on the filer will always be mounted to "/export/home/oracle/dbs" whenever the server boots. For more information see sub-section 3.5. Filer NFS Mount Point.

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

```
# mount /export/home/oracle/dbs
```

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

6.2. Creating the Initialization File

You need to create a database initialization file which places the 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/oracle/dbs/ctl1ORCL.ora, /var/oracle/ctl2ORCL.ora)
```

You can modify the "init.ora" file provided by Oracle for this purpose.

6.3. Starting up the Oracle Instance

The following steps will create a new Oracle9i 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:

   ```
   CONNECT / as sysdba
   STARTUP NOMOUNT
   PFILE=/export/home/oracle/dbs/initORCL.ora
   ```

At that point, the Oracle9i instance should start up. You can now create a database within this instance. Creating a database in the instance is identical to the steps you would normally take. For more information, see the Installation Guide which came with your version of Oracle.

7. Moving an Existing Oracle9i Instance onto a Filer

The following steps will move an existing Oracle9i instance 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 the section section 3 (viz. Infrastructure section) are required in order for this to work.

7.1. Shutting Down the Instance

Shutdown the Oracle9i instance.

```
$ sqlplus

SQL*Plus: Release 9.0.1.0.0 - Production on Mon Sep 24 13:48:43 2001

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Enter user-name: connect/ as sysdba
Enter password:

Connected to:
Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
With the Partitioning option
JServer Release 9.0.1.0.0 - Production
```

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

7.2. Creating the Mount Point

Make changes to the Oracle9i Server machine so that the "/vol/vol1/home/oracle/dbs" directory on the filer will always be mounted to "/export/home/oracle/data" whenever the server boots. For more information see section 3.5. (viz. Filer NFS Mount Point).

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

```
# mount /export/home/oracle/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 Datafiles

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/oracle/data
# cd ..
# rmdir dbs
# ln -s /export/home/oracle/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 9.0.1.0.0 - Production on Mon Sep 24 14:01:22 2001

(c) Copyright 2001 Oracle Corporation. All rights reserved.

SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area    235701300 bytes
Fixed Size                    279604 bytes
Variable Size               167772160 bytes
Database Buffers              67108864 bytes
Redo Buffers                  540672 bytes
Database mounted.
Database opened.
SQL>
```

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

8. Mirroring Certain Oracle Datafiles

If you are running an Oracle9i instance with datafiles on a filer, you must mirror certain Oracle datafiles 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.

First, 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/oracle/dbs/ctl1ORCL.ora, /var/oracle/ctl2ORCL.ora)
```

Then shutdown your instance, copy the control file to both locations, and start it up again.

Second, create mirrored sets of on-line redo log files onto local disk and the filer. For more information on creating mirrored sets of on-line redo log files, see the documentation that came with your version of Oracle8.

Although the Network Appliance filer is a robust, fault-tolerant device, the approach taken in this Technical Report is to absolutely ensure the capability to recover the Oracle9i 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 Oracle9i database with datafiles on a Network Appliance filer.

9. Enabling Database Support on the Filer

It is a good idea to enable the feature of Data ONTAP (Network Appliance filer's operating system software) which supports special error processing when connected to a database server. To do this, enter the following command from the filer's console or a telnet session:

```
vol options vol1 nvfail on
```

This will cause the filer to issue appropriate error messages in the "/etc/messages" file in the case of system failure that might affect the Oracle database. The administrator will learn of these errors either by examining the message logs or by the autosupport email notification feature of the filer.

In particular, this option 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.) outage. Only a failure of the NVRAM card itself should cause it to become invalid. If the contents of 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 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 which 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. Since this occurs before the filer is providing network service, these files will no longer have the same file name as previously. Thus, the applications using them cannot automatically restart (including an Oracle instance which accesses these files).

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/vol1/home/oracle/dbs/ctl1ORCL.dbf
```

to be renamed when the NVRAM failure is detected in version 4.3 of Data ONTAP or earlier would involve creating on the filer the file */etc/nvfail_rename* with the one line:

```
/vol/vol1/home/oracle/dbs/ctl1ORCL.dbf
```

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

```
/vol/vol1/home/oracle/dbs/ctl1ORCL.dbf.nvfail
```

This will prevent the Oracle9i Server from opening the file, and thus make the DBA fully aware of the nvram failure.

10. Caveats

The use of network-attached storage is supported by Oracle only in the context of Network Appliance filers. For more information regarding use of Network Appliance filers with Oracle, see *Using ORACLE with a Multiprotocol Filer* by Bruce Clarke. Further, NetApp has not tested this configuration with any version of UNIX other than Sun Solaris, and has certainly not tested with all of the combinations of hardware and software options available on Solaris. There may be significant differences in your configuration which will alter the procedures necessary to accomplish the objectives outlined in this paper. If you find that any of these procedures do not work in your environment, please contact the author immediately.

11. Revision History

Date	Name	Description
4/05/2001	Sankar Bose	Update
9/24/2001	Jerry Liu	Update