CHAPTER 34

Removing the RAC Option Without Reinstalling

Oracle Real Application Clusters (RAC) can only operate if an instance of Oracle’s own cluster software, which is mandatory starting from Oracle 10g, or a so-called vendor cluster software is available on the same node as the Oracle DBMS instance. This chapter explores the impact of cluster software failure, which in turn prevents RAC instances from opening a database.

Oracle’s cluster software uses a so-called Cluster Registry to save configuration information. It also writes special messages to voting disks to orchestrate cluster actions after an interconnect network failure. In cases of an emergency, such as the loss of all voting disks or the failure of all devices holding copies of the Cluster Registry, Oracle Clusterware cannot be started. In such a severe fault scenario, it might take hours or even days to attach new SAN (Storage Area Network) storage to the system, create logical units (LUNs) in the SAN disk array, and configure zoning to make the LUNs visible to the database server. Oracle Clusterware may cause unjustified node reboots and thus unplanned downtime, although it has matured tremendously since the early days. This might be another reason for disabling RAC. This chapter explains how to use an undocumented make command to remove the RAC option from the oracle executable. Since ASM cannot run at all without Oracle Clusterware infrastructure, a procedure that converts an Oracle Clusterware installation for RAC to a local-only Oracle Clusterware installation for use with single-instance ASM is shown. Thus, a system running RAC can quickly be converted to a system running single instance ORACLE without reinstalling any software with Oracle Universal Installer (OUI) and patching an ORACLE_HOME with OPatch, greatly reducing the downtime incurred by such a severe outage.

Linking ORACLE Software

When ORACLE Server software is installed on a UNIX system by OUI, many programs, including $ORACLE_HOME/bin/oracle, which implements the database kernel, are linked with static and shared libraries on the system. OUI calls the utility make and passes it the makefile $ORACLE_HOME/rdbms/lib/ins_rdbms.mk as an argument. Other makefiles such as $ORACLE_HOME/network/lib/ins_net_server.mk are used to link Oracle Net components and still others to link SQL*Plus, and so on.

Similar steps occur when a patch set (OUI) or interim patch (OPatch) is applied. Most patches modify a static library in $ORACLE_HOME/lib by replacing an object module with a newer version that includes a bug fix. The executable oracle must be relinked to pick up a changed
object module in a static library. For someone who knows how to use the commands ar and make, it is fairly easy to manually apply an interim patch. This is useful when OPatch fails for whatever reason. For example, newer releases of OPatch (starting in versions 1.0.0.0.54 and 10.2.0.x) are able to verify that a new object module (extension .o, e.g., dbsdrv.o) was correctly inserted into a static library (extension .a, e.g., libserver10.a) by extracting the object module and comparing it to the one shipped with the patch. This very reasonable test failed on Solaris 64-bit ORACLE installations, since Solaris 10 pads object files with newline characters (use od -c filename to check this). OPatch complained with a message that said “Archive failed: failed to update” and backed out the interim patch. Setting the environment variable OPATCH_DEBUG=TRUE, which is documented in Oracle Universal Installer and OPatch User's Guide 10g Release 2, revealed that it was not the ar command that was failing, but instead the verification. Meanwhile this issue has been taken care of (see Metalink note 353150.1 for details).

The same approach of exchanging object modules is used to add or remove ORACLE server options. Options can only be purchased with the Enterprise Edition of the ORACLE DBMS. Currently, ten options exist, among them are the following:

- Label Security
- Partitioning
- Real Application Clusters
- Spatial
- Data Mining

The Oracle10g OUI has a bug that causes Data Mining to be installed unconditionally—even when it was deselected on the relevant OUI screen. The approach to add and remove options, which will be presented shortly, may be used to work around this bug.

Which options are installed becomes evident when SQL*Plus is started.

```
$ sqlplus ndebes/secret
SQL*Plus: Release 10.2.0.3.0 - Production on Thu Jul 26 02:21:43 2007
Copyright (c) 1982, 2006, Oracle. All Rights Reserved.
Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, Real Application Clusters, Oracle Label Security and Data Mining options

SQL*Plus takes this information from the view V$OPTION:

SQL> SELECT parameter, value FROM v$option
WHERE parameter IN ('Partitioning', 'Real Application Clusters',
'Oracle Label Security', 'Data Mining')
PARAMETER                           VALUE
---------------------------------------------
Partitioning                        TRUE
Real Application Clusters          TRUE
Oracle Label Security              TRUE
Data Mining                        TRUE

As an option is added or removed by linking, V$OPTION.VALUE becomes TRUE or FALSE.