

MIGRATING YOUR EBUSINESS SUITE SINGLE INSTANCE TO REAL APPLICATION CLUSTERS (RAC)

Ahmed Alomari
Oracle Corporation

INTRODUCTION

This document will describe the steps required to install the EBusiness Suite in an environment which uses Real Application Clusters (RAC). Throughout this document, the term Real Application Clusters (RAC) is synonymous with Oracle Parallel Server (OPS). There are essentially two cases when using RAC with the EBusiness Suite: the fresh install case, or the case whereby the EBusiness Suite is already deployed using a single database instance. The steps to migrate to RAC are essentially the same for both cases. There are two sets of steps: pre-requisite steps which are required before the migration commences, and the migration steps.

PRE-REQUISITE STEPS:

The following steps should be performed prior to the migration of the database to RAC. The pre-requisite steps can be performed at any time prior to the migration to RAC. Both the pre-requisite steps and the RAC migration steps require downtime, hence you should plan accordingly.

STEP 1: COMPLETE THE RAPID INSTALL OF THE EBUSINESS SUITE

If you already have an existing EBusiness Suite runtime environment, you can skip this step. You should complete the Rapid Install and use the standard single database instance installation mode. You should also ensure that the Application modules including Forms and Self-Service are functional. In addition, you should ensure that the Concurrent Managers are active and functional.

STEP 2: CONFIGURE THE SHARED DISK VOLUMES

RAC requires shared disk volumes in order to facilitate the implementation of a shared disk architecture. Using the disk volume manager for your platform such as Veritas, configure the shared disk volumes. In order to minimize the migration time, you should configure a separate shared volume for each database file including database files, redo log files, and control files. All database files must be placed on shared volumes so as to allow access to all the files from all the instances. Although redo log files are private per instance, other instances may need to access the redo log files in the case of recovery. If your database consists of 80 data files, 4 redo log files, and 2 control files, then you should configure at least 86 shared disk volumes. The volume should be sized such that it can contain the original data file. For example, if the existing data file is 1 GB, then the shared volume for that data file should be at least 1 GB in size. In general, you should size the shared volume such that it is larger than the existing data file by several megabytes in order to account for volume manager and raw device overhead.

Once you have completed configuring the shared volumes, bring the shared volumes online, and change the permissions of the shared volumes such that the shared volumes can be read and written by the system account which owns the oracle binary.

STEP 3: INSTALL THE PARALLEL SERVER OPTION

Using the Oracle8i or Oracle9i Universal Installer from the \$ORACLE_HOME on the database server, install the Parallel Server option (if not already installed).

STEP 4: ENSURE THAT THE MAXINSTANCES PARAMETER IS SET PROPERLY

Converting to RAC requires that the database support multiple instances mounting and accessing the same database. Hence, the MAXINSTANCES parameter of the control file should be set to the number of active RAC instances. For example, if configuring a two-node cluster, then MAXINSTANCES should be set to at least two (2). In order to determine the existing value of MAXINSTANCES, run the following query:

```
select type,records_total
from v$controlfile_record_section
where type='REDO THREAD';
```

OUTPUT:

TYPE	RECORDS_TOTAL
REDO THREAD	4

If the value for RECORDS_TOTAL for the REDO THREAD type is greater than or equal to the desired number of RAC instances, then you can skip step 5. Otherwise, follow the instructions in Step 5 in order to increase the value of MAXINSTANCES. In the example output provided above, MAXINSTANCES is set to 4, hence there is no need to recreate the control file.

STEP 5: INCREASE THE MAXINSTANCES PARAMETER

In order to change the control file parameter MAXINSTANCES, you need to recreate the database control file. This can be done via the following steps:

5A) BACKUP THE EXISTING CONTROL FILE

```
myhost> sqlplus "/" as sysdba"
SQL> alter database backup controlfile to trace;
```

5B) LOCATE AND EDIT THE TRACE FILE

The backup control file command will generate a trace file in the directory specified by the init.ora parameter user_dump_dest. Locate the trace file, and edit the trace file in order to create a SQL script which can be used to recreate the control file. You should remove the trace file comments. Change the value for the MAXINSTANCES parameter from 1 to 4, for example.

```
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "OPSPROD" NORESETLOGS NOARCHIVELOG
  MAXLOGFILES 255
  MAXLOGMEMBERS 2
  MAXDATAFILES 1022
  MAXINSTANCES 4
  MAXLOGHISTORY 907
.....
```

5C) SAVE THE CHANGES INTO A SQL SCRIPT

Save the changes made to the trace file in a SQL script named control.sql.

5D) RECREATE THE CONTROL FILE

Shut down all the application services and then the database (in normal mode). Then recreate the control file as follows:

```
myhost> sqlplus "/ as sysdba"
SQL>@control.sql
```

Ensure that the control file has been successfully created, and that the database has opened successfully. Repeat Step 4 from the Pre-requisite steps section in order to confirm that the MAXINSTANCES parameter has been set correctly.

RAC MIGRATION STEPS:

The following steps should be performed following the completion of the pre-requisite steps.

STEP 1: SHUTDOWN THE APPLICATION SERVICES

Prior to migrating to the RAC environment, you should shutdown all the application services such as the Concurrent Manager, Forms Server, Report Server, Discoverer, Self Service (Apache JServ and Apache Mod PL/SQL), etc.. Once the middle-tier services have been successfully shut down, you should shut down the database instance in normal or immediate mode.

STEP 2: COPY THE DATABASE FILES TO THE SHARED VOLUMES

The database should be shutdown prior to copying the files. Copy all the database files to the shared volumes using the OS copy utility. On Solaris, you can use the dd command to copy the data files. For example:

```
prompt> dd if=ont_idx1.dbf of=/dev/vx/rdisk/apps1opsdg/vol25 bs=64k
prompt> dd if=mrp_data1.dbf of=/dev/vx/rdisk/apps1opsdg/vol26 bs=64k
```

STEP 3: LINK TO THE SHARED VOLUMES

After completing the copy of the database files to the shared volumes, you need to ensure that the database now points to the shared volumes. You can either create symbolic links which point to the new shared volumes or rename the file names stored in the database via the command "ALTER DATABASE RENAME FILE." To rename the data files via the ALTER DATABASE RENAME FILE command, the database needs to be started in the exclusive mount mode. After renaming the data files, the database can be opened.

If the ALTER DATABASE RENAME FILE command was used to rename the files, open the database via the command “ALTER DATABASE open” following the completion of the rename. Query v\$datafile, v\$logfile, and v\$tempfile to ensure that the database file names reflect the shared volume names. Then, shutdown the database instance in normal mode.

STEP 4: CONFIGURE THE INIT.ORA

Modify the init.ora file for each instance to include RAC related configuration, such as the instance number and thread number. In addition, the parameter parallel_server should be set to TRUE in order to enable RAC. An example is provided below:

```
instance 1:
    instance_number=1
    thread=1
    parallel_server=true1
    rollback_segments = (rbs1, rbs2, rbs3, rbs4)

instance 2:
    instance_number=2
    thread=2
    parallel_server=true1
    rollback_segments = (rbs5, rbs6, rbs7, rbs8)
```

It is recommended that you assign rollback segments to a particular instance. For example, if your existing database consists of 8 rollback segments (rbs1 - rbs8), you can dedicate four (4) rollback segments to instance #1, and four (4) rollback segments to instance #2.

STEP 5: START INSTANCE #1

Start instance #1 in parallel mode:

```
sqlplus> startup parallel
```

STEP 6: ADD A REDO THREAD FOR INSTANCE #2

Using instance #1, add another redo log thread in order to allow instance #2 to be started.

```
ALTER DATABASE ADD LOGFILE THREAD 2
    'logthr2_1.dbf' size 500M ,
    'logthr2_2' size 500M ;
ALTER DATABASE ENABLE PUBLIC THREAD 2;
```

¹ In Oracle9i, the init.ora parameter parallel_server has been replaced with cluster_database. For RAC environments using Oracle9i, set cluster_database=TRUE.

STEP 7: START INSTANCE #2

Start instance #2 in parallel mode:

```
sqlplus> startup parallel
```

STEP 8: CONFIGURING THE TNSNAMES.ORA FILE

Change the tnsnames.ora files on the middle-tier systems (i.e. database clients) to reflect the RAC environment. For a 2-node cluster system, you should define two separate TNS aliases. Do not use the LOAD_BALANCE option in the tnsnames.ora as this will result in application errors.

STEP 9: CONFIGURING SELF-SERVICE FOR RAC

The Self-Service infrastructure (ICX), by default, uses the database instance name as part of the cookie name if the cookie name has not been explicitly set by the administrator. In the RAC environment, the database instance name can change based on which instance the users were connected. Hence, you should explicitly set the cookie name (SESSION_COOKIE_NAME) to a constant value in the ICX_PARAMETERS table. Update the ICX_PARAMETERS table from the Apps schema using the following example.

```
SQL> update ICX_PARAMETERS
      set SESSION_COOKIE_NAME='OPSPROD1' ;
```

In the example above, the cookie name is set to 'OPSPROD1.' You may specify any value for the session_cookie_name not to exceed 30 characters in length, however, it is recommended that you use a short name (5 characters or less).

STEP 10: START THE APPLICATION SERVICES

Startup the Application servers including the Forms servers, Report Server, Apache (JServ and Mod PL/SQL), Concurrent Manager, etc...

STEP 11: APPLICATION PARTITIONING

In order to direct a certain class of users to a particular instance in the RAC cluster, you can use the Oracle Applications profile option 'Database Instance' (i.e. INSTANCE_PATH) to bind users with a particular responsibility to a particular instance. For example, you can bind Purchasing Forms users to instance A, and Order Management Forms users to instance B using the 'Database Instance' profile option.

For concurrent programs, you can bind concurrent programs to a specific instance by using the Parallel Concurrent Processing (PCP) feature along with specialized managers. For more information on setting up PCP with RAC, refer to the sections "Managing Parallel Concurrent Processing" and "Concurrent Managers and Oracle Parallel Server (OPS)" in the Oracle Applications System Administrator's Guide (Chapter 5).