Certainly! When converting an Oracle Single Instance (SI) database to a RAC One Node database, the pfile (parameter file) will need to be adjusted to include parameters specific to RAC configurations. Here is an example of what the pfile might look like before and after the conversion.

**Before Conversion (Single Instance)**

ini

Copy code

# Single Instance parameter file

db\_name=orcl

instance\_name=orcl

memory\_target=2G

processes=300

audit\_file\_dest=/u01/app/oracle/admin/orcl/adump

control\_files=(/u01/app/oracle/oradata/orcl/control01.ctl, /u01/app/oracle/fast\_recovery\_area/orcl/control02.ctl)

db\_block\_size=8192

db\_create\_file\_dest=/u01/app/oracle/oradata

db\_recovery\_file\_dest=/u01/app/oracle/fast\_recovery\_area

db\_recovery\_file\_dest\_size=20G

diagnostic\_dest=/u01/app/oracle

open\_cursors=300

undo\_tablespace=UNDOTBS1

**After Conversion (RAC One Node)**

ini

Copy code

# RAC One Node parameter file

db\_name=orcl

instance\_name=orcl1

memory\_target=2G

processes=1000

audit\_file\_dest=/u01/app/oracle/admin/orcl/adump

control\_files=(+DATA/orcl/controlfile/current.256.123456789, +RECO/orcl/controlfile/current.256.123456789)

db\_block\_size=8192

db\_create\_file\_dest=+DATA

db\_recovery\_file\_dest=+RECO

db\_recovery\_file\_dest\_size=20G

diagnostic\_dest=/u01/app/oracle

open\_cursors=300

undo\_tablespace=UNDOTBS1

cluster\_database=true

thread=1

instance\_number=1

service\_names=orcl

**Key Changes for RAC One Node**

1. **Instance Parameters**:
   * instance\_name is changed to orcl1 to uniquely identify the RAC One Node instance.
   * processes parameter is increased to handle additional connections typical in RAC environments.
2. **Cluster Parameters**:
   * cluster\_database is set to true to enable RAC.
   * thread and instance\_number parameters are added to uniquely identify the instance in the cluster.
3. **File Locations**:
   * Paths for control\_files, db\_create\_file\_dest, and db\_recovery\_file\_dest are changed to use Oracle ASM (Automatic Storage Management) with disk groups (+DATA and +RECO).

**Additional Parameters for RAC Environments**

For a RAC One Node setup, additional parameters might also be necessary depending on the specific requirements and configuration of the environment:

* cluster\_interconnects: This parameter may be set to specify the interconnect IP addresses used for RAC communication.
* gc\_files\_to\_locks: Specifies the list of files for which Oracle Clusterware manages lock states.

**Example of srvctl commands for RAC One Node**

After adjusting the pfile, you typically use srvctl commands to register and manage the RAC One Node instance:

sh

Copy code

srvctl add instance -db orcl -instance orcl1

srvctl start instance -db orcl -instance orcl1

**Conclusion**

Converting from a Single Instance to RAC One Node involves changes to the parameter file and the management of the database instance using Oracle Clusterware tools. The above example provides a basic template, but additional configurations might be necessary based on the specific requirements and architecture of your Oracle environment.

Do I need to copy this pfile to NodeB and do I need to make any changes on Node B side ?

ChatGPT

Yes, you will need to ensure that the pfile or spfile is available and correctly configured on both nodes of the RAC One Node setup. Here are the detailed steps:

**Step-by-Step Guide**

1. **Copy the PFILE to Node B**:
   * You need to copy the updated pfile from Node A to Node B. Use secure copy (scp) or any other method to transfer the file.

sh

Copy code

scp /path/to/pfile NodeB:/path/to/pfile

1. **Adjust Paths for Node B**:
   * Ensure that any file paths (like those for control files, diagnostic destinations, etc.) are accessible and valid on Node B. If your paths are uniform across nodes using shared storage (like ASM), this step might not require changes.

**Example PFILE for Node B**

Since RAC One Node allows the database to run on any one of the nodes at a time, the pfile usually remains the same across nodes, especially if ASM is used for storage. The spfile will also typically be stored in shared storage, so you do not need different configurations for each node.

**Sample PFILE Copying Process**

1. **Node A PFILE**:

ini

Copy code

db\_name=orcl

instance\_name=orcl1

memory\_target=2G

processes=1000

audit\_file\_dest=/u01/app/oracle/admin/orcl/adump

control\_files=(+DATA/orcl/controlfile/current.256.123456789, +RECO/orcl/controlfile/current.256.123456789)

db\_block\_size=8192

db\_create\_file\_dest=+DATA

db\_recovery\_file\_dest=+RECO

db\_recovery\_file\_dest\_size=20G

diagnostic\_dest=/u01/app/oracle

open\_cursors=300

undo\_tablespace=UNDOTBS1

cluster\_database=true

thread=1

instance\_number=1

service\_names=orcl

1. **Copy to Node B**:

sh

Copy code

scp /u01/app/oracle/product/12.2.0/dbhome\_1/dbs/initorcl.ora oracle@NodeB:/u01/app/oracle/product/12.2.0/dbhome\_1/dbs/

**Use an SPFILE for Easier Management**

In a RAC environment, it's common to use an SPFILE (stored in ASM) to avoid the hassle of managing separate PFILEs on each node.

1. **Create SPFILE from PFILE**:

sql

Copy code

CREATE SPFILE='+DATA/orcl/spfileorcl.ora' FROM PFILE='/u01/app/oracle/product/12.2.0/dbhome\_1/dbs/initorcl.ora';

1. **Start the Database Using SPFILE**:
   * Ensure your pfile points to the spfile location in shared storage.

sh

Copy code

startup pfile='/u01/app/oracle/product/12.2.0/dbhome\_1/dbs/initorcl.ora';

* + The PFILE should contain a reference to the SPFILE:

ini

Copy code

SPFILE='+DATA/orcl/spfileorcl.ora'

1. **Srvctl Configuration**:
   * Configure the RAC One Node database and instance using srvctl:

sh

Copy code

srvctl add database -db orcl -oraclehome /u01/app/oracle/product/12.2.0/dbhome\_1 -spfile '+DATA/orcl/spfileorcl.ora' -dbname orcl

srvctl add instance -db orcl -instance orcl1

srvctl start database -db orcl

**Final Verification**

* Ensure that the database starts correctly on both nodes by switching the instance:

sh

Copy code

srvctl relocate database -db orcl -node NodeB

* Check the status:

sh

Copy code

srvctl status database -db orcl

By following these steps, you ensure that the configuration is consistent across both nodes and that the database can run seamlessly in a RAC One Node environment.

4o

In a RAC One Node environment, you should run the redo log group commands on the node where the RAC One Node database instance is currently running. This is because the commands need to be executed in the context of the running instance to modify the database configuration.

### Steps to Add Redo Log Groups in RAC One Node

1. **Identify the Active Node**: Determine which node the RAC One Node instance is currently running on. You can use the srvctl status command to check this:

sh

Copy code

srvctl status database -db orcl

This command will show you which node the instance is currently running on.

1. **Connect to the Active Node**: Log in to the node where the instance is currently running.
2. **Run the Redo Log Group Commands**: Use SQL\*Plus or another SQL client to connect to the database and add the redo log groups. You can do this locally on the active node.

sh

Copy code

sqlplus / as sysdba

Once connected to SQL\*Plus, run the following commands to add the redo log groups for the new instance (Thread 2):

sql

Copy code

-- Add new redo log groups for the second instance

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 4 ('+DATA/orcl/redo04.log') SIZE 100M;

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 5 ('+DATA/orcl/redo05.log') SIZE 100M;

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 6 ('+DATA/orcl/redo06.log') SIZE 100M;

-- Enable the thread for the second instance

ALTER DATABASE ENABLE THREAD 2;

### Example

Assume the instance is currently running on Node A. You would:

1. Log in to Node A.
2. Connect to SQL\*Plus:

sh

Copy code

sqlplus / as sysdba

1. Execute the redo log group commands:

sql

Copy code

-- Add new redo log groups for the second instance

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 4 ('+DATA/orcl/redo04.log') SIZE 100M;

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 5 ('+DATA/orcl/redo05.log') SIZE 100M;

ALTER DATABASE ADD LOGFILE THREAD 2 GROUP 6 ('+DATA/orcl/redo06.log') SIZE 100M;

-- Enable the thread for the second instance

ALTER DATABASE ENABLE THREAD 2;

### After Adding Redo Log Groups

Once you have added the redo log groups and enabled the thread, you can:

1. **Verify the Redo Log Groups**: Check that the new redo log groups are correctly added.

sql

Copy code

SELECT group#, thread#, status, members FROM v$log;

SELECT \* FROM v$logfile;

1. **Relocate the RAC One Node Instance** (optional): If you want to test or ensure that the configuration is correct, you can relocate the RAC One Node instance to another node.

sh

Copy code

srvctl relocate database -db orcl -node NodeB

1. **Check the Status**: After relocation, verify the status of the database.

sh

Copy code

srvctl status database -db orcl

By following these steps, you ensure that the redo log groups are added correctly and the database configuration is consistent across the RAC One Node environment.