|  |  |  |
| --- | --- | --- |
| **Migrating Databases using Instance Caging to a CDB (文档 ID 1567141.1)** | [IMG_256](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1%26_adf.ctrl-state=sd40kk4r4_72%26_afrLoop=250274384187571)  [转到底部](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571) | IMG_257 |

IMG_258

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In this Document**   |  |  | | --- | --- | |  | [Purpose](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571" \l "REF_PURPOSE) |  |  |  | | --- | --- | |  | [Details](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571" \l "REF_TEXT) |  |  |  | | --- | --- | |  | [Monitoring and Tuning the CDB Plan](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571" \l "aref_section21) |  |  |  | | --- | --- | |  | [Additional Information](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571" \l "aref_section22) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?id=1567141.1&_adf.ctrl-state=sd40kk4r4_72&_afrLoop=250274384187571" \l "REF) |    Purpose Pre-12c, many customers consolidated multiple database instances on a server and used Instance Caging to manage each database instance's CPU usage.  With 12c, a database instance can be converted into a pluggable database (PDB) and consolidated within a CDB.  The CDB Resource Plan can then be used to manage each PDB's CPU usage.  This document shows how to convert the Instance Caging configuration to a CDB resource plan. Details Instance Caging limits the number of CPUs that a database can use at any time.  The number of CPUs is specified by the cpu\_count parameter and Instance Caging is enabled by turning on a resource manager plan.  Instance Caging is described in detail in MOS [Document 1362445.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1567141.1&id=1362445.1).  Suppose that databases BLUE, RED, and YELLOW share a server with 24 CPUs.  Instance Caging is configured as follows.  Note that since the sum of the cpu\_count parameters exceeds the number of CPUs, the server is over-subscribed and some CPU contention between the databases is expected.   |  |  | | --- | --- | | **Database Name** | **cpu\_count** | | BLUE | 16 | | RED | 8 | | YELLOW | 8 |   To create the equivalent CDB Resource Plan, set each PDB's SHARE to its cpu\_count value.  Set each PDB's utilization limit to cpu\_count / num\_cpus.   |  |  |  | | --- | --- | --- | | **PDB Name** | **shares** | **utilization\_limit** | | BLUE | 16 | 66% | | RED | 8 | 33% | | YELLOW | 8 | 33% |   The SHAREs specify the relative amount of CPU each PDB can use.  The UTILIZATION\_LIMIT is a hard limit for the PDB.  If you want to take full advantage of the CPU resources and allow a PDB to use all the CPU when the other PDBs are inactive, then you can remove the UTILIZATION\_LIMIT setting.  This conversion approach works whether the PDBs are originally allocated on the same server or different servers.  If they are on different servers with different CPU types, you may choose to slightly increase the shares for the PDBs on the more powerful servers.  The following script implements this example. It should be run in the ROOT container.  begin    dbms\_resource\_manager.create\_pending\_area;    dbms\_resource\_manager.create\_cdb\_plan(plan => 'mycdb\_plan');    dbms\_resource\_manager.create\_cdb\_plan\_directive(plan => 'mycdb\_plan', pluggable\_database => 'blue', shares => 16, utilization\_limit => 66);    dbms\_resource\_manager.create\_cdb\_plan\_directive(plan => 'mycdb\_plan', pluggable\_database => 'red', shares => 8, utilization\_limit => 33);    dbms\_resource\_manager.create\_cdb\_plan\_directive(plan => 'mycdb\_plan', pluggable\_database => 'yellow', shares => 8, utilization\_limit => 33);    dbms\_resource\_manager.submit\_pending\_area;  end;  /   alter system set resource\_manager\_plan = mycdb\_plan;  Note that CPU\_COUNT does not need to be configured for the CDB.  It should only be set if multiple CDBs are sharing the same server. Monitoring and Tuning the CDB Plan Once a CDB resource plan is deployed, you should monitor the CPU usage and throttling for each PDB and tune the CDB resource plan as needed.  The following query shows the average number of sessions that are running and waiting for CPU for each PDB:  select to\_char(begin\_time, 'HH24:MI'), name, sum(avg\_running\_sessions) avg\_running\_sessions, sum(avg\_waiting\_sessions) avg\_waiting\_sessions from v$rsrcmgrmetric\_history m, v$pdbs p where m.con\_id = p.con\_id group by begin\_time, m.con\_id, name order by begin\_time;  AVG\_RUNNING\_SESSIONS tells you how the CPU is being distributed between the PDBs.  AVG\_WAITING\_SESSIONS tells you which PDBs are being throttled.  If a PDB's AVG\_WAITING\_SESSIONS is high, you can improve its performance by increasing its SHARES and/or UTILIZATION\_LIMIT. Additional Information For more information on Resource Manager, refer to MOS [Document 1339769.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1567141.1&id=1339769.1).   References [NOTE:1339769.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1567141.1&id=1339769.1) - Master Note for Oracle Database Resource Manager |