10g RAC on Linux for PDB - Installation Procedure

This document describes the installation steps for Oracle 10g for the physics database services at CERN. The key software and hardware components are: Oracle 10gR2, RAC, ASM, Linux RHEL 4, dual CPU Intel servers, SAN network, SATA disks in FC arrays (see also https://twiki.cern.ch/twiki/pub/PSSGroup/HAandPerf/Architecture_description.pdf)

OS Setup and Prerequisites

- Check the installation, CDB profiles, kernel version, kernel parameters and other OS installation details.
- Check the public network IPs: **all nodes must be on the same subnet**
- If it's a fresh installation you may need to configure dam (from pdb-backup):
  - `cd ~/dam`
  - `./daminit`
  - `dam add_account oracle@NODE` (if the account is not yet in dam)
  - `dam enable_account oracle@NODE`
  - `dam generate_keys oracle@NODE`
- Check and update info on the pdb_inventory DB
- upload/refresh to the nodes the scripts directory with PDB tools
  - copy (connected to pdb backup) `scp -r $HOME/scripts NODE:`
  - deploy and configure .bashrc (connected to the target NODE as oracle)
    - 64 bit: `cp $HOME/scripts/bashrc_x86_64_sample $HOME/.bashrc; vi .bashrc`
    - 32 bit: `cp $HOME/scripts/bashrc_sample $HOME/.bashrc; vi .bashrc`
  - `source .bashrc`
  - `mkdir $HOME/work $HOME/oracle_binaries`
- Block OS upgrades: `sudo touch /etc/nospma`

Network setup (private interconnect, public IP)

Configure RAC networking on all nodes using a script copied previously from pdb_backup (scripts/rac_net_conf.sh run as oracle on each machine):

- rac_net_conf.sh cluster_name starting_node_number number_of_nodes priv1_network priv2_network or without parameters for interactive mode:
  - ON RAC5,6 * `cd scripts; ./rac_bond_interconnect_conf.sh test1 '601,602,603,...' 172.31.X` **(edit line)**
  - ON RAC2,3,4
    - `cd scripts; ./rac_net_conf.sh test1 415 6 172.31.7 172.31.8` **(edit line)**

- `less /etc/hosts` for cluster interconnect names, virtual ip names, etc
- Check network configs `less /etc/sysconfig/network-scripts/ifcfg-ethX` (X=0,1,2)
  - eth0 is the public interface, should be OK
  - make sure there are **no duplicate IPs** (check at OS level and update the table on pdb inventory with the subnet you want to use)
  - check ifcfg-eth1 and ifcfg-eth2 for the correct IPs and netmasks (use a configured node as an example)

```
# more ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=none
MASTER=bond0
SLAVE=yes

# more ifcfg-eth2
```

```
DEVICE=eth2
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes

# more ifcfg-bond0
DEVICE=bond0
TYPE=Ethernet
MTU=9000
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.31.4.2
NETMASK=255.255.255.0
BROADCAST=172.31.4.255
```

Setup ssh and host equivalence

- Provided that DAM setup (see above) has been configured one can now use simplified ssh setup procedure
  - From **pdb-backup** set up ssh equivalence (**edit last line**):

    ```
    cd ~/scripts
    ./ssh_cluster_setup.sh itrac '601,602,...'
    ```
  - at prompt reply **y** twice to continue
  - check if the last phase returns any error if yes, try again or revert to old procedure (see below)

Deprecated ssh setup procedure

- - On **each node**: `cp $HOME/.ssh/authorized_keys $HOME/.ssh/authorized_keys.old`
    - From **one cluster node** only, set up ssh equivalence:
      `. cd $HOME/scripts; ./sshUserSetup.sh -hosts "host1 host1.cern.ch host2 host2.cern.ch ..." -advanced` **(edit line)**
      - ..at prompt reply **yes** to the first question and **no** to the second
      - ..input the oracle pass multiple times as requested (tip: copy once to the clipboard and paste multiple time)
    - On each node: `cp $HOME/.ssh/authorized_keys $HOME/.ssh/authorized_keys.local`
    - On each node: `cat $HOME/.ssh/authorized_keys.old >>$HOME/.ssh/authorized_keys`

Setup storage: multipathing with device mapper and rawdevices (no asmlib) (the full section to be run as root)

- Identify and prepare the disks/LUNS to by used by CRS and ASM
  - Reload the Qlogic driver to re fresh the disk list
    - IF **RAC3,4,5,6** THEN: `rmmod qla2400 qla2xxx; modprobe qla2400`
    - IF **RAC1,2** THEN: `rmmod qla2300 qla2xxx; modprobe qla2300`
  - **fdisk -l |grep Disk** to list the disks
  - tip: you need to identify the disks (/dev/sd..) with their storage names and LUN (itstor..). For this reason, often best is, to attach one array at a time to configure multipathing.

- on all nodes change ownership to oracle for raw and dm devices
  - `sudo vi /etc/udev/permissions.d/50-udev.permissions`
  - edit the entry raw/*... (change root:root to oracle:ci)
  - edit the entry dm-*... (change root:root to oracle:ci)

- Setup multipathing
  - Generate entries with the script **gen_multipath.py** (in scripts) - copy the generated entries into /etc/multipath.conf header, please check the result file
  - this will generate persistent names in **/dev/mapper** and **/dev/mpath**, note CRS disks and ASM disks have different suffixes
  - **copy** over to the rest of cluster

- start multipathing (one off will need to be executed on all nodes as detailed below)

```
modprobe dm-multipath
modprobe dm-round-robin
chkconfig multipathd on
multipath
```

- **Partition the disks**
- on **RAC5 & 6** after setting up multipathing you can use script that can be found in `scripts/RAC56_stor` called **exec_partall.sh** (all disks need to be attached and configured with multipathing, the script will list storages and ask for confirmation)

```
cd ~/scripts/storage
./exec_partall.sh
```

- on RAC2,3,4 use scripts in old/NEW_ITSTOR_storage_reorg

- On all nodes configure rawdevices
- **in some cases CRS partitions have changed to CRSp1, CRSp2, CRSp3, please check in /dev/mapper, please check if it's your case**

```
vi /etc/sysconfig/rawdevices

/dev/raw/raw1 /dev/mpath/itstorXXX_CRSp1
/dev/raw/raw2 /dev/mpath/itstorXXX_CRSp2
/dev/raw/raw3 /dev/mpath/itstorXXX_CRSp3

/dev/raw/raw11 /dev/mpath/itstorYYY_CRSp1
/dev/raw/raw12 /dev/mpath/itstorYYY_CRSp2

/dev/raw/raw22 /dev/mpath/itstorZZZ_CRSp2
```

- `service rawdevices restart`

- Sync storage config on all cluster nodes
  - sync partition tables by bouncing qla* modules: `sudo multipath -F; sudo /sbin/rmmod qla2400 qla2xxx; sudo /sbin/modprobe qla2400`
  - copy over /etc/multipath.conf and start multipath daemons and rawdevice services on each node

- optionally restart cluster servers: `shutdown -r now`

- Write to netops and ask for network aliases to be used in the tnsnames.ora

Clusterware and RDBMS Installation

Oracle rdbms and ASM will share the same Oracle Home, CRS will need a dedicated home.

Oracle clusterware installation (10.2.0.1 with multipath patch + 10.2.0.4 patchset)

- On all cluster nodes run as root (pconsole in scripts/my_pconsole is a terminal fanout that can be of help for clusters of many nodes):

```
echo "/sbin/modprobe hangcheck-timer" >> /etc/rc.d/rc.local
echo "session    required     pam_limits.so" >> /etc/pam.d/login
/sbin/modprobe hangcheck-timer

mkdir /ORA/dbs00/oracle
chown oracle:ci /ORA/dbs00/oracle
chown oracle:ci /ORA/dbs01/oracle
mkdir /ORA/dbs01/oracle/oraInventory
chown oracle:ci /ORA/dbs01/oracle/oraInventory
echo "inventory_loc=/ORA/dbs01/oracle/oraInventory" > /etc/oraInst.loc
echo "inst_group=ci" >> /etc/oraInst.loc
```

- make sure the rawdevices are 'CLEAN'

```
cd ~/scripts/storage
./clean_CRS_disks.sh
```

- and the equivalent for data disks (the script asks for confirmation):

```
cd ~/scripts/storage
./clean_data_disks.sh
```

- **as oracle** connected to pdb backup copy the relevant files to one target cluster node
- 64 bit installations:
  - `cd $HOME/oracle_binaries/rdbms_102_x86_64`
  - `scp 10201_clusterware_linux_x86_64.cpio.gz TARGET_NODE:$HOME/oracle_binaries`
  - (on each node, mandatory unless cloning) `scp p4679769_10201_Linux-x86-64.zip TARGET_NODE:$HOME/oracle_binaries`
  - (obsolete: 10.2.0.3 patchset) `scp p5337014_10203_Linux-x86-64.zip TARGET_NODE:$HOME/oracle_binaries`
  - (obsolete: on each node, 10.2.0.3-specific work-around) `scp Bug5722352_x86_64_init.cssd TARGET_NODE:$HOME/oracle_binaries`
  - (10.2.0.4 patchset) `scp p6810189_10204_Linux-x86-64.zip TARGET_NODE:$HOME/oracle_binaries`
- 32 bit installations:
  - `cd $HOME/oracle_binaries/rdbms_102_x86`
  - `scp 10201_clusterware_linux32.zip TARGET_NODE:$HOME/oracle_binaries`
  - `scp p5337014_10203_LINUX.zip TARGET_NODE:$HOME/oracle_binaries`
  - `scp p4679769_10201_LINUX.zip TARGET_NODE:$HOME/oracle_binaries`
  - `scp Bug5722352_init.cssd TARGET_NODE:$HOME/oracle_binaries`

- Install from clusterware CD (+ download patch p4679769):
  - (if needed) `DISPLAY=[your_pc_name]:0.0; export DISPLAY;`
  - 64 bit: `cd $HOME/oracle_binaries/; zcat 10201_clusterware_linux_x86_64.cpio.gz| cpio -idmv`
  - 32 bit: `cd $HOME/oracle_binaries/; unzip 10201_clusterware_linux32.zip`
  - `cd clusterware; ./runInstaller`
  - Installation inputs and parameters:
    - ◊ answer 'y' when asked if rootpre.sh has been run by root
    - ◊ set HOME NAME="OraCrs10g"
    - ◊ set ORA_CRS_HOME=/ORA/dbs01/oracle/product/10.2.0/crs
    - ◊ configure cluster and node names
    - ◊ set cluster name = db name
    - ◊ edit cluster nodes: use node_name, alias for the first private interconnect, alias for the vip (see /etc/hosts), note do not specify the domain (.cern.ch). Note, you can also use

a cluster config file.
- 6 raw devices are needed:
  - **OCR:** /dev/raw/raw1, /dev/raw/raw11, take care of using 2 different disk arrays
  - **voting disk:** /dev/raw/raw2, /dev/raw/raw12, /dev/raw/raw22, on 3 different disk arrays
  - NOTE: /dev/raw/raw3 will be used later on for the ASM spfile (not used for CRS)
- start the installation but **do not run root.sh** yet
- apply patch p4679769, as detailed here below, on all nodes after the installation and before running root.sh
  - this is to to allow formating of voting disk and ocr, when using multipathing
  - `cd $HOME/oracle_binaries; unzip p4679769_10201_Linux-x86-64.zip; cd 4679769`
  - `sudo cp $ORA_CRS_HOME/bin/clsfmt.bin $ORA_CRS_HOME/bin/clsfmt.bin.bak; cp clsfmt.bin $ORA_CRS_HOME/bin/clsfmt.bin; sudo chown oracle:ci $ORA_CRS_HOME/bin/clsfmt.bin.bak`
  - repeat on each node
- when prompted, run root.sh on each node: `sudo /ORA/dbs01/oracle/product/10.2.0/crs/root.sh`
- press ok on the oui and wait till **all** 3 post installation steps have run

- apply patchset with runinstaller
  - Stop crs on all nodes `sudo $ORA_CRS_HOME/bin/crsctl stop crs`
  - On one node as oracle:
    - 64 bit: `cd $HOME/oracle_binaries; rm -rf Disk1; unzip p6810189_10204_Linux-x86-64.zip ; cd Disk1; ./runInstaller`
    - (obsolete) 32 bit: unzip p5337014_10203_LINUX.zip; cd Disk1; ./runInstaller
  - Apply the patch, select the relevant crs home and apply the patch on all nodes
  - on all nodes run the postinstall script as root: `sudo /ORA/dbs01/oracle/product/10.2.0/crs/install/root102.sh`

- Apply the CRS bundle #2 for 10.2.0.4 (see installation instructions for JAN09 CPU)

- shutdown all cluster nodes and apply oracle recommended fix for oprocd:
  - **note** make sure the clusterware is down before running this command
  - from one node only: `crsctl set css diagwait 13 -force`
  - restart the clusterware

- (obsolete: 10.2.0.3 only, does not apply to 10.2.0.4) Apply on **each node** the fix for Bug 5722352:
  - this is a bug that causes high write activity on /var/log/messages
  - Stop crs on all nodes as root: `$ORA_CRS_HOME/bin/crsctl stop crs`
  - 64 bit: (on each node) `cd $HOME/oracle_binaries; sudo cp Bug5722352_x86_64_init.cssd /etc/init.d/init.cssd`
  - 32 bit: (on each node) `cd $HOME/oracle_binaries; sudo cp Bug5722352_init.cssd /etc/init.d/init.cssd`
  - restart crs ($ORA_CRS_HOME/bin/crsctl start crs)
  - repeat on next node

RDBMS binaries installation

Cloning (method 2) is the preferred method to deploy Oracle RDBMS installations, for uniformity and speed.

- Method 1 (without cloning) Use Oracle runInstaller to install RAC
  - the installer runs from 1 node only (cluster installation)
  - HOME_NAME=OraDb10g_rdbms
  - ORACLE_HOME=/ORA/dbs01/oracle/product/10.2.0/rdbms
  - install Oracle Enterprise Edition (default)

- choose cluster install on first node only
- Install only the software (no DB creation at this stage)
- Apply patchsets and security patches on the installed node

- **Method 2, cloning**:
  - copy the 'master' tar image from pdb backup (oracle_binaries) to all nodes (unless it is a physical standby installation, in that case take a tar ball of the source Oracle Home and make the necessary changes)
  - Ex: `scp $HOME/oracle_binaries/rdbms_102_x86_64/rdbms_10_2_0_4_with_CPU_JAN09_PDB_BUNDLE_v2.t TARGET:/ORA/dbs01/oracle/product/10.2.0`
  - at the destinations
    - `tar rdbms_10_2_0_4_with_CPU_JAN09_PDB_BUNDLE.tgz`
  - on the new nodes perform the cloning operation
    - `cd $ORACLE_HOME/clone/bin`
    - `perl clone.pl ORACLE_HOME="/ORA/dbs01/oracle/product/10.2.0/rdbms" ORACLE_HOME_NAME="OraDb10g_rdbms" '-O"CLUSTER_NODES={itrXX,itrYY}"' '-O"LOCAL_NODE=itrXX"'` (edit node names)
    - repeat for all new nodes, editing LOCAL_NODE value
    - run root.sh on new nodes, as instructed by clone.pl

- Run **netca** to configure listener.ora for the cluster nodes (only for 10gr1: run vipca before this step)
  - cluster configuration
  - listener name: LISTERNER (each node will have a suffix with the node name automatically)
  - choose the correct non-default port
  - after netca, **vi listener.ora**: remove the EXTPROC entry from listener.ora and use node names instead of IPs
  - **rm tnsnames.ora** (netca creates it only on one node with extproc config that we don't need)

ASM and Database creation

ASM instances and diskgroups creation

- run **dbca**,
  - select: configure **ASM for all nodes**
  - use spfile: raw device /dev/raw/raw3
  - **don't** tune spfile parameters and don't create ASM diskgroups yet
  - click finish and exit dbca
- **post-install actions:**
  - stop asm instances (srvctl stop asm -n ...)
  - sqlsys_ASM and **create pfile='/tmp/pfileASM' from spfile='/dev/raw/raw3';**
  - edit parameters as specified below (after diskgroup creation further editing of asm_diskgroup will be done)
  - sqlsys_ASM and **create spfile='/dev/raw/raw3' from pfile='/tmp/pfileASM';**
- **on all nodes** move the dump directoris: `mkdir -p /ORA/dbs00/oracle/admin; mv /ORA/dbs01/oracle/admin/+ASM /ORA/dbs00/oracle/admin/+ASM; mkdir /ORA/dbs00/oracle/admin/+ASM/adump`
- check the changes and restart ASM (1 instance)

ASM parameters:

```
*.asm_diskgroups='' # Note: will need to be changed again after diskgroups' creation
*.asm_diskstring='/dev/mpath/itstor???_??p?','/dev/mpath/itstor???_?p?'
*.db_cache_size=80M
*.cluster_database=true
*.cluster_database_instances=4
```

```
*.instance_type='asm'
*.large_pool_size=20M
*.asm_power_limit=5
*.processes=100
*.remote_login_passwordfile='exclusive'
*.sga_max_size=200M
*.shared_pool_size=90M
*.audit_file_dest='/ORA/dbs00/oracle/admin/+ASM/adump'
*.user_dump_dest='/ORA/dbs00/oracle/admin/+ASM/udump'
*.background_dump_dest='/ORA/dbs00/oracle/admin/+ASM/bdump'
*.core_dump_dest='/ORA/dbs00/oracle/admin/+ASM/cdump'
+ASM4.instance_number=4
+ASM3.instance_number=3
+ASM1.instance_number=1
+ASM2.instance_number=2
+ASM1.local_listener='LISTENER_+ASM1'
+ASM2.local_listener='LISTENER_+ASM2'
+ASM3.local_listener='LISTENER_+ASM3'
+ASM4.local_listener='LISTENER_+ASM4'
```

- as stated above, **don't use dbca to create diskgroups.**
  - rum **sqlsys_ASM** and @listdisks -> will list details of the disks and diskgroups within sqlplus
  - use the **external partition for DATA** diskgroups, and the internal partition for RECOVERY diskgroups
  - **naming convention** for disk groups: [db_name]_datadg1 and [db_name]_recodg1
  - there should be one failgroup per disk array for the data diskgroup (each failgroup named after disk array name) and only 2 failgroups for the reco diskgroup (named fg1 and fg2)
- create failgroups following these constraints: the recovery area will be used for **disk backups**, the **failure of any 2 disk arrays should minimize the impact on data and recovery areas**
  - note below that fg1 and fg2 are not symmetric between data and recovery diskgorups for that reason
  - note: other configs are possible with more failgroups, for example when using only 3 storage arrays, create 3 FG, one per array.
- Example:

```
create diskgroup test2_datadg1 normal redundancy
failgroup itstor625 disk '/dev/mpath/itstor625_*p1'
failgroup itstor626 disk '/dev/mpath/itstor626_*p1'
failgroup itstor627 disk '/dev/mpath/itstor627_*p1'
failgroup itstor628 disk '/dev/mpath/itstor628_*p1'
failgroup itstor629 disk '/dev/mpath/itstor629_*p1'
failgroup itstor630 disk '/dev/mpath/itstor630_*p1';

create diskgroup test2_recodg1 normal redundancy
failgroup fg1 disk '/dev/mpath/itstor625_*p2'
failgroup fg1 disk '/dev/mpath/itstor626_*p2'
failgroup fg1 disk '/dev/mpath/itstor627_*p2'
failgroup fg2 disk '/dev/mpath/itstor628_*p2'
failgroup fg2 disk '/dev/mpath/itstor629_*p2'
failgroup fg2 disk '/dev/mpath/itstor630_*p2';
```

- For RAC 5 & 6 one can use the script shown below (it generates SQL needed for diskgroup creation), just run the script, the output is self-explanatory:

```
cd ~/scripts/storage
./generate_failgroups.sh cluster_name number_of_storages_for_recovery_only
```

- shutdown all asm instances and change the **asm_diskgroup** parameter with the correct values
  - `srvctl stop asm -n ...`
  - sqlsys_ASM -> `create pfile='/tmp/pfileasm.txt' from spfile='/dev/raw/raw3';`

- `vi /tmp/pfileasm.txt` (example edit asm_diskgroups='TEST2_DATADG1','TEST2_RECODG1')
- sqlsys_ASM -> `create spfile='/dev/raw/raw3' from pfile='/tmp/pfileasm.txt';`
- `srvctl start asm -n ...`
- check with : sqlsys_ASM **@listdisks** and select * from v$asm_diskgroup;'

Database and RAC instances creation

- run dbca to create the DB, post installation steps follow
  - select to create cluster database for all nodes
  - custom database (not from a template)
  - enter DB name with domain name .cern.ch
  - uncheck 'configure for EM flag'
  - input password
  - check 'ASM storage'
  - select the DATA diskgroup created as described above
  - use oracle-managed files
  - specify flash recovery area, created as described above (size 1 TB)
  - choose archivelog if needed
  - uncheck all options (dataming, olap,spatial,EM repository)
  - standard database components: leave JVM,XML, remove intermedia
  - don't tune other parameters yet (leave the defaults) but check block size = 8k, character set = WE8ISO5559P1
  - create database + check 'Generate Database Creation Scripts'
  - **NOTE:** never click twice on the 'java buttons' reaction time can be slow

- fine tune db parameters:
  - sqlsys_DB -> `create pfile='/tmp/initdb.txt' from spfile;`
  - `show parameter spfile`
  - shutdown the DB instances **srvctl stop database -d dbname**
  - edit **vi /tmp/initdb.txt** (see parameter values below)
  - change the dump directories filesystem on **all nodes**: `mv /ORA/dbs01/oracle/admin/[DBNAME] /ORA/dbs00/oracle/admin`
  - sqlsys_DB -> Ex: `create spfile='+TEST2_DATADG1/test2/spfiletest2.ora' from pfile='/tmp/initdb.txt';`
  - check on all nodes in $ORACLE_HOME/dbs that the is no spfile{DBNAME}.ora file (or it will be used instead of the spfile in +ASM)

```
change [DBNAME] with the appropriate value

*.archive_lag_target=4000
*.cluster_database_instances=4
*.cluster_database=TRUE
*.compatible='10.2.0.3' #note do not further increase for 10.2.0.4
*.db_block_size=8192
*.db_cache_advice=OFF # (optional) needed for systems with large memory (quadcores) when disablin
*.db_cache_size=6900000000 # if 16GB of RAM and want to disable ASSM, otherwise blank (unset)
*.shared_pool_size=2569011200 # if 16GB of RAM and want to disable ASSM, otherwise blank (unset)
*.streams_pool_size=600m # unset if the streams are not used.
*.java_pool_size=133554432 # if 16GB of RAM and want to disable ASSM, otherwise blank (unset)
*.large_pool_size=133554432 # if 16GB of RAM and want to disable ASSM, otherwise blank (unset)
*.sga_target=0 # values for quadcores, if you want to disable ASSM. In that case you need to spec
*.sga_max_size=10464788480 #value for 16GB of RAM, must set it if sga_target is blank
*.db_create_file_dest='+[DBNAME]_DATADG1' # customize with datadg name
*.db_files=2000
*.db_domain='cern.ch'
# autotuned in 10.2 -> delete the entry from spfile  for *.db_file_multiblock_read_count
*.db_name=[DBNAME]
*.db_recovery_file_dest='+[DBNAME]_RECODG1'
```

```
*.db_recovery_file_dest_size=6000g
# only if planning to use XDB for ftp *.dispatchers='(PROTOCOL=TCP) (SERVICE=[DBNAME]XDB)'
*.filesystemio_options=setall # in principle not needed on ASM, but we set it anyway
*.global_names=TRUE
*.job_queue_processes=10
*.log_archive_dest_1='LOCATION=USE_DB_RECOVERY_FILE_DEST'
*.log_archive_format='log_%t_%s_%r.arc'
*.log_buffer=10485760
*.open_cursors=300
*.parallel_max_servers=20  # may need tuning, streams uses it, parallel query in principle not ne
*.pga_aggregate_target=3g # value for quadcore 16GB , otherwise set to 1400m for  GB of RAM and t
*.processes=2000 # set to 800 for machines with 4GB of RAM
*.recyclebin=OFF # Set to on when Streams bug is fixed
*.remote_listener='...listener_alias_here....'
*.remote_login_passwordfile='exclusive'
*.resource_limit=TRUE
*.undo_management='AUTO'
*.undo_retention=36000
*.audit_file_dest='/ORA/dbs00/oracle/admin/[DB_NAME]/adump'
*.core_dump_dest='/ORA/dbs00/oracle/admin/[DB_NAME]/cdump'
*.background_dump_dest='/ORA/dbs00/oracle/admin/[DB_NAME]/bdump'
*.user_dump_dest='/ORA/dbs00/oracle/admin/[DB_NAME]/udump'
*.audit_trail='db' # Increase if the full backups are taken more rarely than bi-weekly
*._bct_bitmaps_per_file=24   # when an incremental strategy longer than 8 backups is used
*._job_queue_interval=1 # needed by streams, for streams propagation
#*._high_priority_processes=''  set only for systems with 2 cores (i.e. old itracs), do not use o
*.event='26749 trace name context forever, level 2' #streams propagation perf
#for streams capture system set also event 10868, see metalink Note 551516.1 s
*."_buffered_publisher_flow_control_threshold"=30000  # for streams perf10.2.0.4 only
#only if capture 10.2.0.4 is present *."_capture_publisher_flow_control_threshold"=80000

obsolete params:
# use in 10.20.3 only *.event='26749 trace name context forever, level 2','10867 trace name conte

4 instance-specific parameters, typically set correctly by dbca.
There is one entry per parameter per instance:
instance_number, local_listener, thread, undo_tablespace
```

- **NOTE:** configure **local_listener** even when using port 1521, check also the listener alias in tnsnames.ora, the server name to be used is the **VIP address** with fully qualified name (Ex: ..-v.cern.ch). Edit tnsnames.ora accordingly.

Post Installation

- Check Hugepage memory allocation (if Oracle cannot allocate hugepages it will silently use 'normal' memory)
  - ♦ check hugetbs allocation in the last 3 rows of **more /proc/meminfo**

- Apply catcpu scripts from the latest security patch where relevant. Ex:

```
cd $ORACLE_HOME/rdbms/admin
sqlsys_DB
SQL> select count(*) from dba_objects where status='INVALID';
SQL> @catbundle.sql cpu apply
SQL> @?/rdbms/admin/utlrp.sql
SQL> select count(*) from dba_objects where status='INVALID';
SQL> exit
```

- `ALTER DATABASE SET DEFAULT BIGFILE TABLESPACE;`
- `ALTER DATABASE SET TIME_ZONE = '+02:00';`

- change redo log size, number and multiplexing as appropriate. Ex: add 5 redo groups per thread, no multiplexing, redo size 512m and drop old redologs

- use group numbers =10*thread + seq number (ex: group 11,12,13,14,15 for thread 1 etc)
- specify diskgroup name '+{DB_NAME}_DATADG1' to avoid multiplexing (which is the default)
- Note: if you have >5 nodes, then you may have to run redo_logs.sql script twice (there will be old log 11, 12 (thread# 6), ...), because it will not create some new redo log files. The second time it must be run after dropping old redo logs.

```
SQL> @redo_logs.sql

-drop old redologs
(on all instances) alter system switch logfile;
alter system checkpoint global;
(alternative: alter system archive log all)
alter database drop logfile group 1;
...
```

- change undo and temp tbs size as appropriate
- (optional) revoke from public unneeded privs, such as execute on vulnerable packages

```
revoke execute on sys.lt from public;
revoke execute on dbms_cdc_subscribe from public;
revoke execute on dbms_cdc_isubscribe from public;
revoke execute on sys.utl_tcp from public;
revoke execute on sys.utl_http from public;
revoke execute on sys.utl_smtp from public;
revoke execute on sys.dbms_export_extension from public;
```

- Edit tnsnames.ora
  - local tnsnames in particular the service_name parameter (add .cern.ch where appropriate)
  - afs version

Other post-install actions

- **see post install steps in the DBA wiki**
  - Install EM agent in a separate Oracle_Home
  - setup RAC services and tns_names aliases
  - setup logrotation Logrotation
  - setup cernroles CernRoles
  - setup account monitoring
  - Setup backup (TSM client) BackupSetup
  - add to service and asm monitoring RACmon
  - install purge_old_recyclebin(7) scheduler job
  - install kill_sniped_sessions job
  - see also post-install actions in the 'dba subwiki'

Document change log:

- Jan 2009 - LC added info for CPU JAN09
- Oct 2008 - LC reviewed DB parameter list (minor)
- Jul 2008 - DW updated multipath configuration with scripts
- Mar 2008 - L.C. included new quadcores and bonding
- Jan 2008 - L.C. updated to include 10gR2 on x86_64
- Jan 2007 - D.W. changed NIC installation procedure
- Jan 2007 - L.C. Changed ssh installation and added 10.2.0.3 Bug fixes
- Dec 2006 - L.C. Added device mapper multipath and removed asmlib
- Nov 2006 - L.C. Updated for RHEL4, L.C. Nov 2006
- Apr 2006 - L.C. Major update, revised and tested for 10gR2

- Sep 2005 - L.C. First version, 10gR1 procedure

This topic: PSSGroup > Installation_verbose
Topic revision: r167 - 2009-03-26 - SvetozarKAPUSTA