

10g RAC on RHEL 5 64bit 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 5 64bit, 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.
 - ◆ If RHEL 5 still needs to be installed on the machine please follow instructions here to convert Quattor profiles and properly install OS.
- 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/.bash_profile $HOME/scripts NODE:`
 - ◆ deploy and configure `.bashrc` (connected to the target NODE as oracle): `cp $HOME/scripts/bashrc_x86_64_sample $HOME/.bashrc; vi .bashrc`
 - ◆ `source .bashrc`
 - ◆ `mkdir $HOME/work $HOME/oracle_binaries`
- Block OS upgrades: `sudo touch /etc/nospma`
- Block ncm updates: `sudo touch /etc/noquattor`

Network setup (private interconnect, public IP)

Configure RAC networking on all nodes using a script copied previously from `pdb_backup`:

- as oracle on each node of the cluster run: `script_name cluster_name starting_node_number number_of_nodes priv1_network priv2_network` or without parameters for interactive mode:
 - ◆ ON RAC7 (eth2 and eth3 with bonding used for interconnect): `* cd scripts; ./rac7_bond_interconnect_conf.sh test1 '701,702,703,...' 172.31.X (edit line)`
 - ◆ ON RAC5 and 6 (eth1 and eth2 with bonding used for interconnect): `* cd scripts; ./rac_bond_interconnect_conf.sh test1 '601,602,603,...' 172.31.X (edit line)`
 - ◆ ON RAC3 and 4 (no bonding): `* 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-bond0`, `ifcfg-eth1`, `ifcfg-eth2` and `ifcfg-eth3` for the correct IPs and netmasks (use a configured node as an example)

```
# more ifcfg-eth2
DEVICE=eth2
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=none
```

```

MASTER=bond0
SLAVE=yes

# more ifcfg-eth3
DEVICE=eth3
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 '701,702,...'

```

◇ at prompt reply **y** twice to continue

◇ ignore error during the check phase.

- ◆ on all nodes as oracle create a `.ssh/config` file with the following contents: **### FIXME:** to be quattorized

```

Host *
IdentityFile ~/.ssh/identity

```

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 be used by CRS and ASM
 - ◆ on all nodes reload the Qlogic driver to refresh the disk list: `sudo rmmod qla2xxx; sudo modprobe qla2xxx`
 - ◆ **sudo fdisk -l |grep Disk** to list the disks
- on all nodes change ownership to oracle for `/dev/dm` and `/dev/mapper` devices
 - ◆ create and populate the `/etc/udev/rules.d/39-oraclemultipath.rules` file with the following contents:

```

sudo vi /etc/udev/rules.d/39-oraclemultipath.rules
-----
# multipath wants the devmaps presented as meaningful device names
# so name them after their devmap name
SUBSYSTEM!="block", GOTO="end_mpath"
KERNEL!="dm-[0-9]*", ACTION=="add", PROGRAM=="/bin/bash -c '/sbin/lsmode | /bin/grep
KERNEL!="dm-[0-9]*", GOTO="end_mpath"
PROGRAM!="/sbin/mpath_wait %M %m", GOTO="end_mpath"
ACTION=="add", RUN+="/sbin/dmsetup ls --target multipath --exec '/sbin/kpartx -a -p

```

InstallationProcedureRHEL5 < PSSGroup < TWiki

```
PROGRAM=="/sbin/dmsetup ls --target multipath --exec /bin/basename -j %M -m %m", RES
PROGRAM!=" /bin/bash -c '/sbin/dmsetup info -c --noheadings -j %M -m %m | /bin/grep -
PROGRAM=="/sbin/dmsetup ls --target linear --exec /bin/basename -j %M -m %m", NAME=
GOTO="end_mpath"
LABEL="oracle_rules"
RESULT=="itstor*p1", OWNER="oracle", GROUP="dba", MODE="660", RUN+="/bin/chown oracl
RESULT=="itstor*p2", OWNER="oracle", GROUP="dba", MODE="660", RUN+="/bin/chown oracl
RESULT=="itstor*CRSp1", OWNER="oracle", GROUP="oinstall", MODE="660", RUN+="/bin/chd
RESULT=="itstor*CRSp2", OWNER="oracle", GROUP="oinstall", MODE="660", RUN+="/bin/chd
RESULT=="itstor*CRSp3", OWNER="oracle", GROUP="dba", MODE="660", RUN+="/bin/chown or
OPTIONS="last_rule"
LABEL="end_mpath"
```

• Setup multipathing

- ◆ As oracle on one of the nodes generate entries with the script **gen_multipath.py** (in scripts) - copy the generated entries into `/etc/multipath.conf` header, please check the result file

```
gen_multipath.py > /tmp/multipath.conf
sudo mv /tmp/multipath.conf /etc/multipath.conf
```

- ◆ 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. On each remaining cluster nodes do:

```
sudo scp oracle@SOURCE:/etc/multipath.conf /etc/multipath.conf
```

• On all nodes start multipathing

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

• From one of the cluster nodes partition the disks:

- ◆ on **RAC7** use : `cd ~/scripts/storage; sudo ./exec_partall_rac7.sh`
- ◆ on **RAC5 & 6** use : `cd ~/scripts/storage; sudo ./exec_partall.sh`
- ◆ on **RAC3 & 4** use : `cd ~/scripts/storage; sudo ./exec_partall_rac34.sh`

• Reboot cluster nodes

- Use web page <https://network.cern.ch/sc/fcgi/sc.fcgi?Action=Main> to assign DNS aliases

Clusterware and RDBMS Installation

Oracle rdbms and ASM will share the same Oracle Home, CRS will need a dedicated home. For installing both CRS and RDMB/ASM home use cloning.

Oracle clusterware installation (10.2.0.4 patchset + recommended patch bundles)

- 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):

```
sudo bash
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
```

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

InstallationProcedureRHEL5 < PSSGroup < TWiki

```
chown oracle:dba /ORA/dbs00/oracle
chown oracle:oinstall /ORA/dbs01/oracle
mkdir /ORA/dbs01/oracle/oraInventory
chown oracle:oinstall /ORA/dbs01/oracle/oraInventory
mkdir -p /ORA/dbs01/oracle/product/10.2.0
chown oracle:oinstall /ORA/dbs01/oracle/product/10.2.0
echo "inventory_loc=/ORA/dbs01/oracle/oraInventory" > /etc/oraInst.loc
echo "inst_group=oinstall" >> /etc/oraInst.loc
exit
```

- make sure the CRS disks 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 tarball to all nodes of the cluster

```
cd $HOME/oracle_binaries/rdbms_102_x86_64
scp crs_10_2_0_4_BUNDLE3.tgz TARGET_NODE:/ORA/dbs01/oracle/product/10.2.0
```

- On all nodes of the cluster as oracle untar and clone the CRS home:

- ◆ cd /ORA/dbs01/oracle/product/10.2.0; sudo tar xzpf crs_10_2_0_4_BUNDLE3.tgz
- ◆ sudo chown -R oracle:oinstall \$ORA_CRS_HOME
- ◆ run a script that will generate cloning commands:
 - ◇ on RAC7 run: rac7_crs_clone.sh
 - ◇ on RAC5 & 6 run: rac56_crs_clone.sh
 - ◇ on RAC3 & 4 run: rac34_crs_clone.sh
 - ◇ specify cluster name and numbers of the nodes.
 - ◇ this script prepares the command line for cloning and 'patch' to be run on all nodes in the following step
 - ◇ note the post installation step takes care of modifying netmask for VIPS, a subtle but very important detail
- ◆ On each cluster node run cloning command generated by the script above:

```
cd $ORA_CRS_HOME/clone/bin
perl clone.pl ORACLE_HOME=...
# Example:
perl clone.pl ORACLE_HOME="/ORA/dbs01/oracle/product/10.2.0/crs" ORACLE_HOME_NAME="C
```

- ◇ The meaning of different parameters is the following:

- Cluster name: s_clustername (e.g. int11r)
- Primary copy of OCR (use a block device): s_ocrpartitionlocation (e.g. /dev/mapper/itstor305_CRSp1)
- Secondary copy of OCR (use a block device): s_ocrMirrorLocation (e.g. /dev/mapper/itstor306_CRSp1)
- Voting disk locations (use block devices): s_votingdisklocation, s_OcrVdiskMirror1RetVal, s_VdiskMirror2RetVal
- List of public, private and virtual names of all cluster nodes: s_l_tableList (e.g. {itrac305:int11r-priv1-1:itrac305-v:N:Y,itrac306:int11r-priv1-2:itrac3

InstallationProcedureRHEL5 < PSSGroup < TWiki

· Public and private networks: `ret_PrivIntrList` (e.g.
{eth0:137.138.0.0:1,eth1:172.31.41.0:2,eth2:172.31.42.0:2}

- ◆ **Before running** `root.sh`, on each node run `sed` command generated by the `racX_crs_clone.sh` script:

```
# Example
sed -r "s#^(CRS_NODEVIPS=')\.*#\1itrac305/itrac305-v,itrac306/itrac306-v'#g" $ORA_CRS_HOME/
mv $ORA_CRS_HOME/install/paramfile.crs $ORA_CRS_HOME/install/paramfile.crs.orig; mv
sed -r "s#^(CRS_NODEVIPS=')\.*#\1itrac305/itrac305-v/255.255.0.0/eth0,itrac306/itrac306-v/255.255.0.0/eth0'#g" $ORA_CRS_HOME/
mv $ORA_CRS_HOME/install/rootconfig $ORA_CRS_HOME/install/rootconfig.orig; mv $ORA_CRS_HOME/install/rootconfig $ORA_CRS_HOME/install/rootconfig.orig
```

- ◆ **After completing the steps above** run `root.sh` on each node but **only on one node at a time**:
`sudo /ORA/dbs01/oracle/product/10.2.0/crs/root.sh`
- ◆ Once the clusterware is running on one node of the cluster run post-installation script (as stated in the output of the `racX_crs_clone.sh` script):

```
cd $ORA_CRS_HOME/cfgtoollogs
./configToolAllCommands
```

- Verify with the `crsstat.sh` script that the cluster is running and nodeapps are up.

RDBMS binaries installation

- 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_APR09_PDB_BUNDLE_v2_RHEL5.tar.gz`
`TARGET:/ORA/dbs01/oracle/product/10.2.0`
- at the destinations
 - ◆ `cd /ORA/dbs01/oracle/product/10.2.0; tar xfpz rdbms_10_2_0_4_with_CPU_APR09_PDB_BUNDLE_v2_RHEL5.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

- **NOTE:** Since we want to use a block device for the ASM spfile (and OUI does not support that) we need to create an ASM instance manually.
- on one of the nodes prepare a parameter file (e.g. `/tmp/initASM.ora`) with contents similar to the one below:

InstallationProcedureRHEL5 < PSSGroup < TWiki

```
*.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'
```

- Choose a block device to be used to store the spfile for ASM (e.g. /dev/mapper/itstorXXX_CRSp3)
- Create the spfile on the selected block device:

```
sqlsys_ASM
create spfile='/dev/mapper/itstorXXX_CRSp3' from pfile='/tmp/initASM.ora';
```

- On each node of the cluster create a pfile for ASM in the default location pointing to the spfile:

```
cd $ORACLE_HOME/dbs; echo "SPFILE='/dev/mapper/itstorXXX_CRSp3'" > init+ASM${INST_NUM}.ora
```

- On each node create an ASM password file:

```
$ORACLE_HOME/bin/orapwd file=$ORACLE_HOME/dbs/orapw+ASM${INST_NUM} password=XXXXXXXX
```

- On each node create dump directories:

```
mkdir -p /ORA/dbs00/oracle/admin/+ASM/adump
mkdir -p /ORA/dbs00/oracle/admin/+ASM/bdump
mkdir -p /ORA/dbs00/oracle/admin/+ASM/cdump
mkdir -p /ORA/dbs00/oracle/admin/+ASM/udump
```

- On each node modify `tnsnames.ora` file defining aliases used for the `LOCAL_LISTENER` parameter.

Example:

```
LISTENER_+ASM1 =
  (ADDRESS = (PROTOCOL = TCP) (HOST = it...) (PORT = ..))

LISTENER_+ASM2 =
  (ADDRESS = (PROTOCOL = TCP) (HOST = it...) (PORT = ..))
```

- On one node add ASM instances to the clusterware and try to start them up:

```
# for each ASM instance do:
srvctl add asm -n itracXXX -i +ASMX -o $ORACLE_HOME
srvctl start asm -n itracXXX
```

- Create diskgroups (from one of the nodes):

- ◆ run `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 diskgroups 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/mapper/itstorXXX_CRSp3';`
 - ◆ `vi /tmp/pfileasm.txt` (example edit `asm_diskgroups='TEST2_DATADG1','TEST2_RECODG1'`)
 - ◆ `sqlsys_ASM -> create spfile='/dev/mapper/itstorXXX_CRSp3' 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 there 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 disabling
*.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 specify
*.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 needed
*.pga_aggregate_target=3g # value for quadcore 16GB, otherwise set to 1400m for 4GB 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'
```


InstallationProcedureRHEL5 < PSSGroup < TWiki

```
*.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, do not use on quadcore systems  
*.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:

- JUNE 2009 - JW,DW,LC, major update for RHEL5
- 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 > InstallationProcedureRHEL5

Topic revision: r29 - 2009-07-14 - LucaCanali



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback