System Capacity Test - RunBook TEMPLATE

# Create RunBook Document

**DO NOT EDIT TEMPLATE DOCUMENT. Save as a copy using following format**

**System capacity test - RunBook [dbname run 1] [date YYYYMMDD run 1] [dbname run 2] [date YYYYMMDD run 2]**

Example:

"System Capacity Test - RunBook DTCDUAT1 20170324 DTCDUAT 20170418"

# Create Log Document

**DO NOT EDIT TEMPLATE DOCUMENT. Save as a copy using following format**

Open "**System capacity test - Log TEMPLATE - READONLY**" document and save it as   
**System capacity test - Log [dbname run 1] [date YYYYMMDD run 1] [dbname run 2] [date YYYYMMDD run 2]**

Example:  
  
"System capacity test - Log DTCDUAT1 20170324 DTCDUAT 20170418"

# Replace Variables - Use [Ctrl+H]

Run1 - Baseline Portion

|  |  |
| --- | --- |
| 19990131 | New Date of test  YYYYMMDD |
| 1999-01-31 01:00:00 | New Timestamp of beginning of the test YYYY-MM-DD HH24:MI:SS |
| 1999-01-31 02:00:00 | New Timestamp of end of the test YYYY-MM-DD HH24:MI:SS |
| Jan/31/1999 01:00:00 | New Timestamp of beginning of the test Mon/DD/YYYY HH24:MI:SS |
| Jan/31/1999 02:00:00 | New Timestamp of end of the test Mon/DD/YYYY HH24:MI:SS |
| Jan 31 01:00:00 1999 | New Timestamp of beginning of the test Mon DD HH24:MI:SS YYYY |
| Jan 31 02:00:00 1999 | New Timestamp of end of the test Mon DD HH24:MI:SS YYYY |
| 01/31/99 | Remember to modify time portion for ASH report That report requires specific time when unusual activity happened  New Date of test  MM/DD/YY |
| 01-31-99 | MM-DD-YY |
| 9931010100 | YYMMDDHHMI  Start time of the test |
| DBSID99 | With database name |
| stha9999 | With node where database resides |
| 9999999999 | with Current database's DBID  export ORACLE\_SID=DBSID99  export ORAENV\_ASK=NO  . oraenv  sqlplus / as sysdba<<EOF  select dbid from v\$database;  exit  EOF |

Run 2 - New test

|  |  |
| --- | --- |
| 20010228 | New Date of test  YYYYMMDD |
| 2001-02-28 11:00:00 | New Timestamp of beginning of the test YYYY-MM-DD HH24:MI:SS |
| 2001-02-28 12:00:00 | New Timestamp of end of the test YYYY-MM-DD HH24:MI:SS |
| Feb/28/2001 11:00:00 | New Timestamp of beginning of the test Mon/DD/YYYY HH24:MI:SS |
| Feb/28/2001 12:00:00 | New Timestamp of end of the test Mon/DD/YYYY HH24:MI:SS |
| Feb 28 11:00:00 2001 | New Timestamp of beginning of the test Mon DD HH24:MI:SS YYYY |
| Feb 28 12:00:00 2001 | New Timestamp of end of the test Mon DD HH24:MI:SS YYYY |
| 02/28/01 | Remember to modify time portion for ASH report That report requires specific time when unusual activity happened  New Date of test  MM/DD/YY |
| 02-28-01 | MM-DD-YY |
| 0102281100 | YYMMDDHHMI  Start time of the test |
| DBSID01 | With database name |
| stha1111 | With node where database resides |
| 1111111111 | with Current database's DBID  sqlplus / as sysdba<<EOF  select dbid from v\$database;  exit  EOF |

# References

Best Practices: Proactive Data Collection for Performance Issues (Doc ID 1477599.1)  
ORAchk - Health Checks for the Oracle Stack (Doc ID 1268927.2)  
OSWatcher Analyzer User Guide (Doc ID 461053.1)  
OSWatcher (Includes: [Video]) (Doc ID 301137.1)  
OSW : What Is The OSWATCHER Effect On The Server Performance And What Are The Alternative Commands? (Doc ID 946107.1)  
How to Tell if the I/O of the Database is Slow (Doc ID 1275596.1)  
How to Calculate the Number of IOPS and Throughput of a Database (Doc ID 2206831.1)  
Multiple Log Writers in 12c Ceising Enabling and Disabling of Adaptive Scalable Log Writer Workers Which Ceise High 'log file sync' Wait Event (Doc ID 2174075.1)   
Adaptive Switching Between Log Write Methods can Ceise 'log file sync' Waits (Doc ID 1462942.1)  
Recommendation for the Real Application Cluster Interconnect and Jumbo Frames (Doc ID 341788.1)  
Analysis of Active Session History (Ash) Online and Offline (Doc ID 243132.1)  
ashdump\* scripts and post-load processing of MMNL traces (Doc ID 555303.1)

<https://orainternals.wordpress.com/2013/06/12/dude-where-is-my-redo/>  
<http://dbastreet.com/blog/?p=646> Automate ASH Reports

<https://flashdba.com/database/useful-scripts/awr-generator/> Automate AWR Reports

<http://www.oracle.com/technetwork/database/manageability/emx-activerep-1965979.html> Active reports

<https://oracle-base.com/articles/misc/sql-trace-10046-trcsess-and-tkprof> tracing

<http://www.juliandyke.com/Diagnostics/Trace/EnablingTrace.php> Tracing individual sqlid

# TODO

Create AWR SQL Report for Each SQL in TOP SQL Report by Elapsed Time <- NOT COMPLETED YET

blog.tanelpoder.com/archives/

tpt\_public.zip

Install RDA

|  |
| --- |
| unzip -o /dbabin/dbatools/rda/p21769913\_81517314\_AIX64-5L.zip -d /su02/dbatools  cd /su02/dbatools/rda  rda.sh |

# Scheduling Capacity Test

IMPORTANT!!!

Start time of Performance/Capacity tests MUST be synced with beginning of AWR snapshots.

Example: If snapshots start @HH24:00 or HH24:30, then Performance/Capacity tests MUST start @HH24:00 or HH24:30

# Pre-Capacity Validation

## ORACHK report

|  |
| --- |
| cd /dbabin/dbatools/orachk  ./orachk -a |

Download latest and run "orachk -a"  
  
<https://updates.oracle.com/Orion/Services/download/orachk.zip?aru=20593189&patch_file=orachk.zip>

ORAchk - Health Checks for the Oracle Stack (Doc ID 1268927.2)

## Check JumboFrames (RAC Only)

|  |
| --- |
| [dbarkin@stha8n0dk ~]$ ifconfig eth1  eth1 Link encap:Ethernet HWaddr 00:21:F6:01:52:2E  inet addr:192.168.1.198 Bcast:192.168.1.255 Mask:255.255.255.0  inet6 addr: fe80::221:f6ff:fe01:522e/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:9000 Metric:1  RX packets:597896738 errors:0 dropped:15476 overruns:0 frame:0  TX packets:566494874 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:562236239686 (523.6 GiB) TX bytes:464635044691 (432.7 GiB) |

|  |
| --- |
| ping -M do -s 8972 stha8n04n-ha.iad.ca.inet  ping -M do -s 8972 stha8n04m-ha.iad.ca.inet  traceroute -F stha8n04n-ha.iad.ca.inet 9000 traceroute to node02-priv (10.10.10.2), 30 hops max, 9000 byte packets  traceroute -F stha8n04n-ha.iad.ca.inet 9001    [dbarkin@stha8n0dk ~]$ ping -M do -s 8972 stha8n0dl-ha.iad.ca.inet  PING stha8n0dl-ha.iad.ca.inet (192.168.1.199) 8972(9000) bytes of data.  8980 bytes from stha8n0dl-ha.iad.ca.inet (192.168.1.199): icmp\_seq=1 ttl=64 time=0.329 ms  8980 bytes from stha8n0dl-ha.iad.ca.inet (192.168.1.199): icmp\_seq=2 ttl=64 time=0.345 ms  [dbarkin@stha8n0dk ~]$ traceroute -F stha8n0dl-ha.iad.ca.inet 9000  traceroute to stha8n0dl-ha.iad.ca.inet (192.168.1.199), 30 hops max, 9000 byte packets  1 stha8n0dl-ha.iad.ca.inet (192.168.1.199) 0.282 ms 0.223 ms 0.176 ms  [dbarkin@stha8n0dk ~]$ traceroute -F stha8n0dl-ha.iad.ca.inet 9001  traceroute to stha8n0dl-ha.iad.ca.inet (192.168.1.199), 30 hops max, 9001 byte packets  1 stha8n0dk-ha.iad.ca.inet (192.168.1.198) 0.035 ms !F-0 0.012 ms !F-0 0.012 ms !F-0  [dbarkin@stha8n0dk ~]$ |

Expected output Node 1

|  |
| --- |
| [dbarkin@stha8n04m ~]$ ping -M do -s 8972 stha8n04n-ha.iad.ca.inet  PING stha8n04n-ha.iad.ca.inet (192.168.1.145) 8972(9000) bytes of data.  8980 bytes from stha8n04n-ha.iad.ca.inet (192.168.1.145): icmp\_seq=1 ttl=64 time=0.297 ms  8980 bytes from stha8n04n-ha.iad.ca.inet (192.168.1.145): icmp\_seq=2 ttl=64 time=0.300 ms  8980 bytes from stha8n04n-ha.iad.ca.inet (192.168.1.145): icmp\_seq=3 ttl=64 time=0.334 ms  8980 bytes from stha8n04n-ha.iad.ca.inet (192.168.1.145): icmp\_seq=4 ttl=64 time=0.306 ms  8980 bytes from stha8n04n-ha.iad.ca.inet (192.168.1.145): icmp\_seq=5 ttl=64 time=0.365 ms  ^C  --- stha8n04n-ha.iad.ca.inet ping statistics ---  5 packets transmitted, 5 received, 0% packet loss, time 4250ms  rtt min/avg/max/mdev = 0.297/0.320/0.365/0.030 ms  [dbarkin@stha8n04m ~]$ traceroute -F stha8n04n-ha.iad.ca.inet 9000  traceroute to stha8n04n-ha.iad.ca.inet (192.168.1.145), 30 hops max, 9000 byte packets  1 stha8n04n-ha.iad.ca.inet (192.168.1.145) 0.317 ms 0.272 ms 0.232 ms  [dbarkin@stha8n04m ~]$ traceroute -F stha8n04n-ha.iad.ca.inet 9001  traceroute to stha8n04n-ha.iad.ca.inet (192.168.1.145), 30 hops max, 9001 byte packets  1 stha8n04m-ha.iad.ca.inet (192.168.1.146) 0.029 ms !F-0 0.013 ms !F-0 0.009 ms !F-0  [dbarkin@stha8n04m ~]$ |

Expected output Node 2

|  |
| --- |
| [oracle@stha8n04n ~]$ ping -M do -s 8972 stha8n04m-ha.iad.ca.inet  PING stha8n04m-ha.iad.ca.inet (192.168.1.146) 8972(9000) bytes of data.  8980 bytes from stha8n04m-ha.iad.ca.inet (192.168.1.146): icmp\_seq=1 ttl=64 time=0.501 ms  8980 bytes from stha8n04m-ha.iad.ca.inet (192.168.1.146): icmp\_seq=2 ttl=64 time=9.44 ms  ^C  --- stha8n04m-ha.iad.ca.inet ping statistics ---  2 packets transmitted, 2 received, 0% packet loss, time 1570ms  rtt min/avg/max/mdev = 0.501/4.971/9.442/4.471 ms  [oracle@stha8n04n ~]$ traceroute -F stha8n04m-ha.iad.ca.inet 9000  traceroute to stha8n04m-ha.iad.ca.inet (192.168.1.146), 30 hops max, 9000 byte packets  1 stha8n04m-ha.iad.ca.inet (192.168.1.146) 0.860 ms 0.838 ms 0.823 ms  [oracle@stha8n04n ~]$ traceroute -F stha8n04m-ha.iad.ca.inet 9001  traceroute to stha8n04m-ha.iad.ca.inet (192.168.1.146), 30 hops max, 9001 byte packets  1 stha8n04n-ha.iad.ca.inet (192.168.1.145) 0.019 ms !F-0 0.007 ms !F-0 0.005 ms !F-0  [oracle@stha8n04n ~]$ |

## Validate Tools

### TFA Analyzer

Make sure TFA analyzer is installed. Follow TFA Install/Upgrade Guide.

[https://teamsites.iad.ca.inet/ifc/DBA/Oracle Documents/Troubleshooting/Common/TFA/Install or Upgrade TFA 12c.docx](https://teamsites.iad.ca.inet/ifc/DBA/Oracle%20Documents/Troubleshooting/Common/TFA/Install%20or%20Upgrade%20TFA%2012c.docx)

### OSWatcher

Make sure oswatcher during capacity test is run from TFA home.

Stop OSWatcher

Both nodes

|  |
| --- |
| /su02/dbatools/oswbb/stopOSWbb.sh  ps -ef|grep osw  ps -ef|grep OSW  exit  #Make sure you exit from the shell |

Start OSWatcher - Both nodes

|  |
| --- |
| if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  $TFA\_HOME/bin/tfactl print status  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  echo ERROR! NO TFA INSTALLED!  return 1  fi  export PATH=$PATH:$TFA\_HOME/bin    tfactl toolstatus  tfactl stop oswbb  tfactl start oswbb  sleep 10  tfactl toolstatus  ps -ef|grep oswbb |

### Proc Watcher (Optional)

Start Proc Watcher - Both nodes

|  |
| --- |
| if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  else  echo ERROR! NO TFA INSTALLED!  return 1  fi  export PATH=$PATH:$TFA\_HOME/bin  tfactl print status  tfactl toolstatus  tfactl prw stop  tfactl prw start  sleep 10  tfactl toolstatus |

if required modify   
/su01/app/oracle/tfa/repository/suptools/prw/oracle/prwinit.ora

# Pre-Capacity Activities

## Disable Backups

|  |
| --- |
| /dbabin/common/backup2/RMAN\_Backup.ksh DBtoTDPO ALLDB Arc 2  egrep -v '^[[:blank:]]\*#|^[[:blank:]]\*$|\+ASM\*' /etc/oratab|awk -F ":" '{print $1}'|while read LINE  do  echo Disabling backup for $LINE  echo You should see "Database its already in the Ignore File" output. This is expected.  echo If you don\'t see it\, check if backup was already disabled  /dbabin/common/backup2/add\_db\_to\_ignorelist.sh DISABLE\_BACKUP $(hostname) $LINE  /dbabin/common/backup2/add\_db\_to\_ignorelist.sh DISABLE\_BACKUP $(hostname) $LINE  :  done |

## Stop OEM Agents on the nodes

Setup blackouts and stop OEM agents on the nodes.

Run this on Node 1 only

|  |
| --- |
| export ORACLE\_SID=$(grep ^\+ASM /etc/oratab|awk -F ":" '{print $1}')  export ORAENV\_ASK=NO  . oraenv  . /dbabin/dbatools/emcli/emclienv  /dbabin/dbatools/emcli\_dba\_prod/emcli logout  emclia\_info=`perl -e 'print pack "H\*","4c6b386e56682133"'`  /dbabin/dbatools/emcli\_dba\_prod/emcli login -username=EMCLIA -password="$emclia\_info"  unset emclia\_info  /dbabin/dbatools/emcli\_dba\_prod/emcli status  /dbabin/dbatools/emcli\_dba\_prod/emcli sync  export ORA\_CLU=$($ORACLE\_HOME/bin/olsnodes|wc -l)  if [ $ORA\_CLU -eq 0 ]; then  export EMTARGET=$(hostname)".iad.ca.inet:host"  else  export EMTARGET=$($ORACLE\_HOME/bin/cemutlo -n)":cluster"  fi  print $EMTARGET  /dbabin/dbatools/emcli\_dba\_prod/emcli delete\_blackout -name="$EMTARGET capacity test"  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackout\_targets -name="$EMTARGET capacity test"  /dbabin/dbatools/emcli\_dba\_prod/emcli create\_blackout \  -name="$EMTARGET capacity test" \  -add\_targets="$EMTARGET"\  -reason="DB: Database Emergency" -description="$EMTARGET capacity test" \  -propagate\_targets \  -schedule="frequency:once;duration:05:00"  print $EMTARGET  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackout\_targets -name="$EMTARGET capacity test"  print $EMTARGET  if [ $ORA\_CLU -eq 0 ]; then  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackouts -target="$(hostname).iad.ca.inet:host"  else  $ORACLE\_HOME/bin/olsnodes|while read READ\_LINE  do  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackouts -target="${READ\_LINE}.iad.ca.inet:host"  :  done  fi |

Run this on Both nodes

|  |
| --- |
| /su01/app/oracle/cc/agent/agent\_inst/bin/emctl stop agent |

## Disable Automation

Main reason to avoid truncation of trace files due automation log purge

Ref: [https://teamsites.iad.ca.inet/ifc/DBA/Oracle Documents/Automation/Documentation/Exclude a server from the automation.doc](https://teamsites.iad.ca.inet/ifc/DBA/Oracle%20Documents/Automation/Documentation/Exclude%20a%20server%20from%20the%20automation.doc)

|  |  |
| --- | --- |
| Connect on stha24045    cd /oracle\_automation\_nonprod/config  vi odb\_AutomationServerLists.cfg  search for servers participating in capacity testing  Comment the line for the server to be excluded   |  | | --- | | #PREP stha9999 NOALIAS STANDALONE ORA\_9|ORA\_10|ORA\_11|ORA\_12 | |

## Set Proper parameters <- OPTIONAL (AVOID rebooting DB)

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee ${ORACLE\_SID}\_pretest\_parameters.log  show parameter cpu\_count  show parameter "\_use\_single\_log\_writer"  alter system reset cpu\_count scope=spfile;  ALTER SYSTEM SET "\_use\_single\_log\_writer"=TRUE SCOPE=SPFILE;  exit  EOF  srvctl stop database -d DBSID99  srvctl start database -d DBSID99  sqlplus / as sysdba<<EOF  audit CREATE SESSION;  exit  EOF |

## Set statistics level to ALL

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee ${ORACLE\_SID}\_pretest\_parameters.log  alter system set statistics\_level = all;  exit  EOF |

## Verify Load Balancing Service (Savers specific)

|  |
| --- |
| srvctl status service -db DBSID99 -service DBSID99\_HTTP1.INTACT.NET  srvctl status service -db DBSID99 -service DBSID99\_HTTP2.INTACT.NET |

Expected result

|  |
| --- |
| oracle@stha8n0dk:DBSID991(/home/oracle) $ srvctl status service -db DBSID99 -service DBSID99\_HTTP1.INTACT.NET  Service DBSID99\_HTTP1.INTACT.NET is running on instance(s) DBSID991  oracle@stha8n0dk:DBSID991(/home/oracle) $  oracle@stha8n0dk:DBSID991(/home/oracle) $ srvctl status service -db DBSID99 -service DBSID99\_HTTP2.INTACT.NET  Service DBSID99\_HTTP2.INTACT.NET is running on instance(s) DBSID992  oracle@stha8n0dk:DBSID991(/home/oracle) $ |

Fix only if services are not running on appropriate instances

|  |
| --- |
| srvctl stop service -db DBSID99 -service DBSID99\_HTTP2.INTACT.NET  srvctl stop service -db DBSID99 -service DBSID99\_HTTP1.INTACT.NET  srvctl start service -db DBSID99 -service DBSID99\_HTTP2.INTACT.NET  srvctl start service -db DBSID99 -service DBSID99\_HTTP1.INTACT.NET |

## Modify AWR retention

|  |
| --- |
| sqlplus / as sysdba  begin  dbms\_workload\_repository.modify\_snapshot\_settings(  topnsql =>100  );  end;  /  begin DBMS\_WORKLOAD\_REPOSITORY.MODIFY\_SNAPSHOT\_SETTINGS(retention=>0,interval=>30); end;  /  exit |

## Verify AWR information

|  |
| --- |
| sqlplus / as sysdba @?/rdbms/admin/awrinfo.sql |

## Setup Trace (Optional)

|  |
| --- |
| sqlplus / as sysdba<<EOF  EXEC DBMS\_MONITOR.serv\_mod\_act\_trace\_enable(service\_name=><PUT SERVICE NAME HERE>, module\_name=>'pmdtm@stha7n0gx (TNS V1-V3)', waits=>TRUE, binds=>TRUE);  col PRIMARY\_ID format a30  col QUALIFIER\_ID1 format a30  col QUALIFIER\_ID2 format a30  set lin 150  select TRACE\_TYPE,PRIMARY\_ID,QUALIFIER\_ID1,QUALIFIER\_ID2,WAITS,BINDS from dba\_enabled\_traces;  exit  EOF  cd /su01/app/oracle/diag/rdbms/DBSID99/DBSID99/trace/  #http://www.juliandyke.com/Diagnostics/Trace/EnablingTrace.php  #ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=3s1yukp05bzg6|aca4xvmz0rzup] bind=true, wait=true';  #ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=3s1yukp05bzg6|aca4xvmz0rzup] off';  #ALTER SYSTEM SET EVENTS 'sql\_trace bind=true,wait=true'; |

## Send email (Sample email)

|  |
| --- |
| Please inform and communicated to all interested parties.  It will be CL Fleets' performance test/baseline today starting 11:30 AM between 12:00 noon for one and half hour in DBSID99 (stha24541)  Please do not run any jobs on DBSID99 or the server stha24541 and don't use any tools for online database monitoring.  No connections to EISAPERF from AD/Support/DBA teams.  No OEM connections to the database as well. OEM agent will be down during the baseline test.  RMAN backups are disabled on stha24541 during the test |

# Post-Capacity Activities 1 of 2

## Enable Backups

|  |
| --- |
| egrep -v '^[[:blank:]]\*#|^[[:blank:]]\*$|\+ASM\*' /etc/oratab|awk -F ":" '{print $1}'|while read LINE  do  echo Enabling backup for $LINE  /dbabin/common/backup2/add\_db\_to\_ignorelist.sh ENABLE\_BACKUP $(hostname) $LINE  :  done  /dbabin/common/backup2/RMAN\_Backup.ksh DBtoTDPO ALLDB Arc 2 |

## Set statistics level to TYPICAL

|  |
| --- |
| sqlplus / as sysdba<<EOF  alter system set statistics\_level = typical;  exit  EOF |

## Start OEM Agents on the nodes

Stop blackout and start OEM agents on the nodes.

Run this on Both nodes

|  |
| --- |
| /su01/app/oracle/cc/agent/agent\_inst/bin/emctl start agent |

Run this on Node 1 only

|  |
| --- |
| . /dbabin/dbatools/emcli/emclienv  /dbabin/dbatools/emcli\_dba\_prod/emcli logout  emclia\_info=`perl -e 'print pack "H\*","4c6b386e56682133"'`  /dbabin/dbatools/emcli\_dba\_prod/emcli login -username=EMCLIA -password="$emclia\_info"  unset emclia\_info  /dbabin/dbatools/emcli\_dba\_prod/emcli status  /dbabin/dbatools/emcli\_dba\_prod/emcli sync  export ORA\_CLU=$($ORACLE\_HOME/bin/olsnodes|wc -l)  if [ $ORA\_CLU -eq 0 ]; then  export EMTARGET=$(hostname)".iad.ca.inet:host"  else  export EMTARGET=$($ORACLE\_HOME/bin/cemutlo -n)":cluster"  fi  print $EMTARGET  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackout\_targets -name="$EMTARGET capacity test"  print $EMTARGET  if [ $ORA\_CLU -eq 0 ]; then  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackouts -target="$(hostname).iad.ca.inet:host"  else  $ORACLE\_HOME/bin/olsnodes|while read READ\_LINE  do  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackouts -target="${READ\_LINE}.iad.ca.inet:host"  :  done  fi  print $EMTARGET  /dbabin/dbatools/emcli\_dba\_prod/emcli stop\_blackout -name="$EMTARGET capacity test"  /dbabin/dbatools/emcli\_dba\_prod/emcli get\_blackout\_targets -name="$EMTARGET capacity test"  /dbabin/dbatools/emcli\_dba\_prod/emcli delete\_blackout -name="$EMTARGET capacity test"  /dbabin/dbatools/emcli\_dba\_prod/emcli logout |

## Remove Trace (Optional)

|  |
| --- |
| sqlplus / as sysdba<<EOF  EXEC DBMS\_MONITOR.serv\_mod\_act\_trace\_disable(service\_name=><PUT SERVICE NAME HERE>, module\_name=>'pmdtm@stha7n0gx (TNS V1-V3)');  col PRIMARY\_ID format a30  col QUALIFIER\_ID1 format a30  col QUALIFIER\_ID2 format a30  set lin 150  select TRACE\_TYPE,PRIMARY\_ID,QUALIFIER\_ID1,QUALIFIER\_ID2,WAITS,BINDS from dba\_enabled\_traces;  exit  EOF  cd /su01/app/oracle/diag/rdbms/DBSID99/DBSID99/trace/  # ALTER SYSTEM SET EVENTS 'sql\_trace off'; |

# Information Collection RUN1

## Record <R01-LOG-01> Timestamp of the test

Navigate to R01-LOG-01 section of Log document and paste there timestamps of the test

|  |  |
| --- | --- |
| From | To |
| 1999-01-31 01:00:00 | 1999-01-31 02:00:00 |

## Create Directories to hold diagnostics data

Execute this on All nodes

|  |
| --- |
| mkdir -p /home/oracle/perftest/19990131  mkdir -p /home/oracle/perftest/19990131/$(hostname)  mkdir -p /home/oracle/perftest/19990131/$(hostname)/collectionlogs  mkdir -p /home/oracle/perftest/19990131/$(hostname)/tfa  mkdir -p /home/oracle/perftest/19990131/$(hostname)/db/  mkdir -p /home/oracle/perftest/19990131/$(hostname)/db/ash  mkdir -p /home/oracle/perftest/19990131/$(hostname)/db/awr  cd /home/oracle/perftest/19990131 |

## Immediate Collection

### ASH V$ACTIVE\_SESSION\_HISTORY offline dump

Ref: Analysis of Active Session History (Ash) Online and Offline (Doc ID 243132.1)

ashdump\* scripts and post-load processing of MMNL traces (Doc ID 555303.1)

Run this immediately on all RAC instances after the test.

|  |
| --- |
| cd /home/oracle/perftest/19990131 sqlplus / as sysdba<<EOF>/home/oracle/perftest/19990131/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_times.log  col mintime format A25  col maxtime format A25  col minutes format 999  SELECT to\_char(MIN(SAMPLE\_TIME),'YYYY-MM-DD HH24:MI:SS') as mintime,  to\_char(Max(SAMPLE\_TIME),'YYYY-MM-DD HH24:MI:SS') as maxtime FROM V\$ACTIVE\_SESSION\_HISTORY;  with diff as (select systimestamp - TIMESTAMP'1999-01-31 01:00:00' as diff from dual)  SELECT extract(day from diff.diff)\*24\*60+extract(hour from diff.diff)\*60+extract(minute from diff.diff) as minutes FROM diff;  exit  EOF  cat /home/oracle/perftest/19990131/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_times.log |

In the output of the script, check for "Minutes" column. You will use that number to collect ASH offline dump.

|  |
| --- |
| SQL> SQL> SQL> SQL> 2  MINTIME MAXTIME  ------------------------- -------------------------  2017-02-07 11:32:44 2017-02-09 19:11:58  SQL> 2  MINUTES  -------  192 |

**DON'T EDIT TEXT BELOW!**

**Copy it to notepad and change the [**oradebug dump ashdump ####**] with the number of minutes from query above**

|  |
| --- |
| cd /home/oracle/perftest/19990131  sqlplus / as sysdba<<EOF|tee /home/oracle/perftest/19990131/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log  oradebug setmypid  oradebug unlimit  oradebug tracefile\_name  REM number below from minutes column  REM REPLACE NUMBER with number from minutes column from QUERY ABOVE (${ORACLE\_SID}\_ash\_dump\_times.log)  oradebug dump ashdump ####;  exit  EOF  cat /home/oracle/perftest/19990131/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log |

Copy produced file to diagnostic directory

All nodes

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/ash    #Get the filename from ASH collection trace file  cp $(egrep '^SQL>[ ].\*\.trc$' \  /home/oracle/perftest/19990131/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log|sed -e 's/^SQL>[ ]//g') \  $(hostname)\_${ORACLE\_SID}\_ASH\_data.trc  ls -l  #Copy ASH file to network location as well  cp $(hostname)\_${ORACLE\_SID}\_ASH\_data.trc /nfsrwora/12c\_migrate/awrdata/ash/ |

### Record <R01-LOG-02> from OEM ASH Graph on both nodes to Log Document

Navigate to R01-LOG-02 section of Log document and paste there screenshot from Cloud Control ASH Analytics window. Use "Custom" button to specify correct timeframe.

Use at least two dimensions:

* SQLID by Wait Class
* SQL Plan Hash Value by Wait Class
* Do Load Map in Advanced mode

### Extract PerfHUB Report (12c Only)

Use report level ALL

Please enter report level [typical]: all

Run this on all instances of RAC

|  |
| --- |
| cd /home/oracle/perftest/20010131/$(hostname)/db/awr  sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/perfhubrpt.sql |

### Extract Real-Time Active ADDM Report (12c Only)

Change default ADDM range

|  |
| --- |
| vi $ORACLE\_HOME/rdbms/admin/rtaddmrpti.sql  change  define default\_report\_duration = 60;  to  define default\_report\_duration = 1440; |

**Identify ADDM real-time snapshots**

Run this on first instance of RAC only

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/awr sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/rtaddmrpt.sql |

**Check output**

|  |
| --- |
| DB Id REPORT\_ID TIME trigger\_ceise impact  ---------- --------- -------------------- ------------------------- ----------  2654834759 2893 09/02/2017 16:30:51 High Load 3.49  2654834759 2902 09/02/2017 17:21:23 High Load 3.01 |

**Extract ADDM real-time snapshot**

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/awr  sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/rtaddmrpt.sql |

File should use following naming convention:

[active\_rt]\_ADDM\_[YYYYMMDD]\_[HHMMSS]\_[REPORT ID].html

example:

active\_rt\_ADDM\_19990131\_163051\_2893.html

active\_rt\_ADDM\_19990131\_172123\_2902.html

### Collect ASH info

Review ASH Graph, Real Time ADDM, Perf HUB reports.

identify times when the system was busiest. Use CPU, I/O, Redo generation, unusual events to identify busy time.

Script below will use 10min intervals and produce ASH report for various wait classes for 2 instances (RAC).

**DON'T EDIT TEXT BELOW!**

Copy text into notepad and modify Dates and Times.

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/awr  export TIMES\_TMP="times.txt"  >$TIMES\_TMP  echo "01/31/99 01:00" >> $TIMES\_TMP  echo "01/31/99 01:10" >> $TIMES\_TMP  echo "01/31/99 01:20" >> $TIMES\_TMP  echo "01/31/99 01:30" >> $TIMES\_TMP  echo "01/31/99 01:40" >> $TIMES\_TMP  echo "01/31/99 01:50" >> $TIMES\_TMP  export WAITCLASS\_TMP="waits.txt"  >$WAITCLASS\_TMP  echo CPU >> $WAITCLASS\_TMP  echo Cluster >> $WAITCLASS\_TMP  echo Application >> $WAITCLASS\_TMP  echo Concurrency >> $WAITCLASS\_TMP  echo Other >> $WAITCLASS\_TMP  echo User\_I\_O >> $WAITCLASS\_TMP  echo System\_I\_O >> $WAITCLASS\_TMP  ash\_collection\_rac()  {  V\_INSTANCE="$1"  if [[ $2 = "ALL" ]]; then  V\_WAIT\_CLASS=""  else  V\_WAIT\_CLASS="$2"  fi  V\_EVENTTIME="$3"  V\_EVENTTIME\_FILE="$4"  echo 1 $V\_INSTANCE  echo 2 $V\_WAIT\_CLASS  echo 3 $V\_EVENTTIME  echo 4 $V\_EVENTTIME\_FILE  sqlplus / as sysdba<<EOF  define wait\_class='$V\_WAIT\_CLASS';  define eventdatetime\_param='$V\_EVENTTIME'  define eventdatetime\_file='$V\_EVENTTIME\_FILE'  define dbid='';  define inst\_num='$V\_INSTANCE';  define report\_type = '';  define slot\_width = '';  define target\_sql\_id = '';  define target\_wait\_class = '&&wait\_class';  define target\_session\_id = '';  define target\_service\_hash = '';  define target\_module\_name = '';  define target\_action\_name = '';  define target\_client\_id = '';  define target\_plsql\_entry = '';  define target\_container = '';  define duration='10';  define begin\_time='&&eventdatetime\_param';  define report\_name = ashrpt\_&&inst\_num.\_&&eventdatetime\_file.\_&&wait\_class..html  @?/rdbms/admin/ashrpti.sql  exit  EOF  }  cat $TIMES\_TMP | while read TIMES\_LINE  do  FILE\_EXT=$(echo $TIMES\_LINE|sed -e 's/[/:]//g' -e 's/[ ]/\_/g')  ash\_collection\_rac 1 "ALL" "$TIMES\_LINE" "$FILE\_EXT"  ash\_collection\_rac 2 "ALL" "$TIMES\_LINE" "$FILE\_EXT"  cat $WAITCLASS\_TMP | while read WAITCLASS\_LINE  do  echo $WAITCLASS\_LINE $TIMES\_LINE $FILE\_EXT  ash\_collection\_rac 1 "$WAITCLASS\_LINE" "$TIMES\_LINE" "$FILE\_EXT"  ash\_collection\_rac 2 "$WAITCLASS\_LINE" "$TIMES\_LINE" "$FILE\_EXT"  :  done  :  done |

## Next Day Collection

### Load ASH Dump file into permanent table on AWRRPT database (OPTIONAL)

Move the files from original servers (all members of RAC cluster)

Run this on AWRRPT node (stha8n083)

|  |
| --- |
| mkdir -p /home/oracle/perftest/19990131  mkdir /home/oracle/perftest/19990131/db/awr  cd /home/oracle/perftest/19990131  rm -f ashdump\*  cp /dbabin/dbatools/diagscript/ashdump/12c/\* .  ls -l /nfsrwora/12c\_migrate/awrdata/ash/\*ASH\_data.trc |

|  |
| --- |
| ASH\_FILE="PUT REQUIRED FILE FILE " |

|  |
| --- |
| ls -1 $ASH\_FILE 2>/dev/null|while read line  do  echo processing file $line  ./ashdump\_loader $line  echo done processing file $line  :  done  ls -l \*.bad  echo Verify if you had any records in the bad file |

### Rename Table to a table to contain timestamp format

|  |
| --- |
| export frm=19990131  sqlplus / as sysdba<<EOF  alter table ashdump.ashdump rename to ashdump${frm};  exit  EOF |

### Collect lfsdiag.sql Information (log sync)

On first node only

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/awr  sqlplus -s / as sysdba @/dbabin/dbatools/diagscript/lfsdiag.sql |

### Record <R01-LOG-03> Current AWR snapshots

Run script below. Select only information about snapshots and database DBID info. Don't run complete report.

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/db/awr  sqlplus / as sysdba @?/rdbms/admin/awrrpt.sql |

Navigate to R01-LOG-03 section of Log document and paste there AWR snapshots for RUN1 start and finish times.

### Create local AWR report

Login to tested database node and run command to generate local AWR reports

You will need to specify 4 parameters:

* DBID
* Database name (if RAC not SID but a database name)
* First Snapshot in the test
* Last snapshot in the test

source code <https://github.com/dbarkin/awr_auto_reports>

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #BEGIN\_SNAPID# and #END\_SNAPID# with relevant snapshot IDs from R02-LOG-03 section of Log document.

|  |
| --- |
| unalias cp  cd /home/oracle/perftest/19990131/$(hostname)/db/awr  cp /dbabin/dbatools/diagscript/awr\_auto\_reports/\* .  . generate\_local\_awr.sh  generate\_local\_awr 9999999999 DBSID99 758 762 |

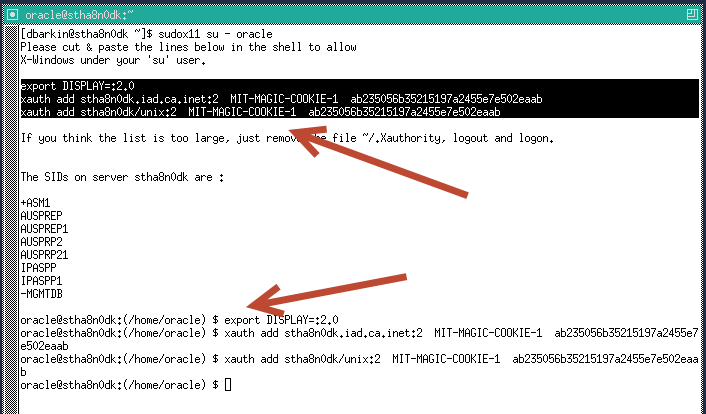
### Start VNC on ALL RAC nodes (required to generate proper graphs using OSWatcher)

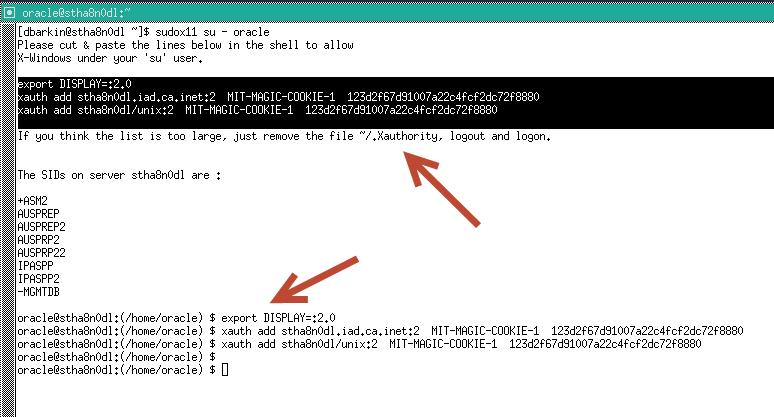
|  |
| --- |
| [dbarkin@stha8n04n ~]$ vncserver :2  VNC(R) Server 5.2.3 (r8648) 64-bit (Jan 28 2015 15:47:49)  Copyright (C) 2002-2015 RealVNC Ltd.  VNC is a registered trademark of RealVNC Ltd in the U.S. and in other  countries.  Protected by UK patent 2481870; US patent 8760366.  See http://www.realvnc.com for information on VNC.  For third party acknowledgements see:  http://www.realvnc.com/products/vnc/documentation/5.2/acknowledgements.txt  Generating private key...done  xauth: creating new authority file /home/dbarkin/.Xauthority  If a desktop environment fails to load for this virtual desktop, please see:  http://www.realvnc.com/kb-345  Running applications in /etc/vnc/xstartup  VNC Server catchphrase: "Spark Joel Kansas. Owner pump sonic."  signature: c7-3b-69-df-64-34-c8-6a  Log file is /home/dbarkin/.vnc/stha8n04n:2.log  New desktop is stha8n04n:2 (10.177.121.74:2)  [dbarkin@stha8n04n ~]$ |

### Open VNC xterminal from your laptop using vncviewer on All RAC Nodes - Required for OSWatcher

|  |  |  |
| --- | --- | --- |
| open VNC xterminal from your laptop using vncviewer on BOTH Nodes   |  |  | | --- | --- | |  |  | |

Make sure you export x11 variables on both nodes





### Run generic TFA diagnostics collection

Run this on all nodes

|  |
| --- |
| export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/19990131/$(hostname)/tfa  >$(hostname)\_tfa\_summary.txt  tfactl summary |tee -a $(hostname)\_tfa\_summary.txt  tfactl events |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp db -from "Jan/31/1999 01:00:00" -to "Jan/31/1999 02:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp os -from "Jan/31/1999 01:00:00" -to "Jan/31/1999 02:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp crs -from "Jan/31/1999 01:00:00" -to "Jan/31/1999 02:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp asm -from "Jan/31/1999 01:00:00" -to "Jan/31/1999 02:00:00" |tee -a $(hostname)\_tfa\_summary.txt  cat $(hostname)\_tfa\_summary.txt  ls -l $(hostname)\_tfa\_summary.txt  #REPLACE Dates with relevant! |

Run this on all nodes

|  |
| --- |
| export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/19990131/$(hostname)/tfa tfactl diagcollect -all -from "Jan/31/1999 01:00:00" -to "Jan/31/1999 02:00:00"|tee /home/oracle/perftest/19990131/$(hostname)/collectionlogs/tfactl\_diagcollect.log  #REPLACE Dates with relevant! |

### Save generic TFA diagnostics collection

Run this on all RAC nodes

|  |
| --- |
| cd /home/oracle/perftest/19990131/$(hostname)/tfa  cp $(ls -1 /su01/app/oracle/tfa/repository/$(ls -1rt /su01/app/oracle/tfa/repository/|tail -1)/\*.zip) .  ls -l \*.zip |

### Run OSW TFA Analysis and GIF graph generation

IMPORTANT!!!

From VNC Session connected to oracle account with `sudox11 su - orace` from xterminal with DISPLAY set and `xauth add` commands

Run this on all RAC nodes

|  |
| --- |
| export DISPLAY=localhost:2  export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/19990131/$(hostname)/tfa  tfactl oswbb -6 -7 -8 -B Jan 31 01:00:00 1999 -E Jan 31 02:00:00 1999 -S|\  tee /home/oracle/perftest/19990131/$(hostname)/collectionlogs/tfactl\_osw\_gif.log |

### Save OSW TFA Analysis

Archive last OSW TFA collection on all nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/analysis/  cp $(ls -1rt|tail -1) /home/oracle/perftest/19990131/$(hostname)/tfa/$(hostname)\_osw\_analysis.txt  cd /home/oracle/perftest/19990131/$(hostname)/tfa  ls -l $(hostname)\_osw\_analysis.txt |

Save OSW TFA GIF Graphs

Identify gif files produced on both nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/gif/ ls -lrt|tail |tee /home/oracle/perftest/19990131/$(hostname)/collectionlogs/$(hostname)\_osw\_gif\_graphs.log |

Archive information on all nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/gif/  cp -r $(ls -1rt|tail -1) /home/oracle/perftest/19990131/$(hostname)/tfa/osw\_gif  cd /home/oracle/perftest/19990131/$(hostname)/tfa  ls -l |

### OS Diagnostic Archive

Run this on all RAC nodes

|  |
| --- |
| cd /home/oracle/perftest/20010228  zip -r ../20010228\_OS\_$(hostname).zip \*  zip -sf ../20010228\_OS\_$(hostname).zip |

## AWR Import

### Create Directory Structure on AWRRPT node

Connect to the node which has AWRRPT database (stha8n083 as of 19990131)

|  |
| --- |
| <stha9999> here is the node where the performance test run  mkdir -p /home/oracle/perftest/19990131/stha9999  mkdir -p /home/oracle/perftest/19990131/stha9999/db/awr |

### Create AWR Data directory on Source Database

Run this on the node where performance test was performed (<stha9999>)

|  |
| --- |
| mkdir -p /nfsrwora/12c\_migrate/awrdata sqlplus / as sysdba <<EOF|tee ${ORACLE\_SID}\_AWR\_datapump\_directoris.log  select \* from global\_name;  create or replace directory AWRDATA as '/nfsrwora/12c\_migrate/awrdata';  create or replace directory AWRLOG as '/nfsrwora/12c\_migrate/awrdata';  set lin 120  column directory\_path format A60  column directory\_name format A30  select directory\_name,directory\_path from dba\_directories where directory\_name like 'AWR%';  exit  EOF |

### Extract AWRs from Source DB

Open "System Capacity - Log" file and Refer to <R01-LOG-03> for Current AWR snapshots

Run this on the node where performance test was performed (<stha9999>)

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #BEGIN\_SNAPID# and #END\_SNAPID# with relevant snapshot IDs from R02-LOG-03 section of Log document.

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee $(date +%Y%m%d%H%M)\_AWR\_load.log  define b\_snap = #BEGIN\_SNAPID#;  define e\_snap = #END\_SNAPID#;  define num\_days = 1;  define db\_name = 'DBSID99';  define dbid = 9999999999;  define directory\_name='AWRDATA';  define begin\_snap = &b\_snap;  define end\_snap = &e\_snap;  column rdate new\_val file\_date  select to\_char(sysdate,'YYMMDD')rdate from dual;  define file\_name=&&db\_name.\_&&begin\_snap.\_&&end\_snap.\_&&file\_date  @$ORACLE\_HOME/rdbms/admin/awrextr.sql  exit  EOF |

### Record <R01-LOG-04> AWR Export Output in the log file

Record AWR Export Output in the "System Capacity Test - Log" file in section <R01-LOG-04>

### Import AWR to AWR Archive database

Connect to the node which has AWRRPT database (stha8n083 as of 19990131)

List required datapump file

|  |
| --- |
| cd /nfsrwora/12c\_migrate/awrdata  ls -lrt \*.dmp|tail |

Export variable with datapump file without .dmp extention

Important: replace NEWDBID!

Important: replace DMPFILE!

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #DMPFILE# with relevant DUMPFILE from "System Capacity Test - Log" file in section <R01-LOG-04>

|  |
| --- |
| . oraenv  AWRRPT  export DMPFILE=#DMPFILE#  # Look for ^^^ space. Remove its  **# Make sure there is no DMP extention^^^^^^**  unset NEWDBID  export NEWDBID=<<<YYMMDDHH24MI>>> of test data Maximum 10 numbers  #Example  export NEWDBID=9931010100 |

Import data

|  |  |
| --- | --- |
| import\_awr(){  export LOG\_FILE=$(date +%Y%m%d%H%M)\_AWR\_load.log  echo IMPORTANT! It will be new DBID $NEWDBID  echo IMPORTANT! Record old DBID and new DBID  sqlplus / as sysdba<<EOF|tee $LOG\_FILE  define directory\_name='AWRDATA';  define schema\_name='AWR\_STAGE';  define file\_name='$DMPFILE';  define default\_tablespace='AWR\_DATA';  define temporary\_tablespace='TEMP';  @$ORACLE\_HOME/rdbms/admin/awrload.sql ${NEWDBID}  exec DBMS\_WORKLOAD\_REPOSITORY.MODIFY\_SNAPSHOT\_SETTINGS( retention=>0, dbid=>${NEWDBID});  exit  EOF  cat $LOG\_FILE  }  if [ -z "$DMPFILE" ]; then  echo "Error! DMPFILE is not set "  else  [[ ! -z "$NEWDBID" ]] && import\_awr || echo "Error! NEWDBID is not set "  fi |  |

**ONLY In Case you need to clean-up failed import**

Identify snapshots and dbid to cleanup

|  |
| --- |
| . oraenv  AWRRPT  sqlplus -s / as sysdba<<EOF  set verify off echo off feed off  set lin 1000  set pagesize 60  column MIN\_SNAP format 99990  column MAX\_SNAP format 99990  column LAST\_INTERVAL format A25  column FIRST\_INTERVAL format A25  column DB\_NAME format A10  column HOST\_NAME format A10 WRAP  column PLATFORM\_NAME format A10 WRAP  select s.DBID, i.db\_name,i.host\_name,substr(i.platform\_name,1,10) platform\_name, min(SNAP\_ID) MIN\_SNAP, max(SNAP\_ID) MAX\_SNAP, min(BEGIN\_INTERVAL\_TIME) FIRST\_INTERVAL, max(BEGIN\_INTERVAL\_TIME) LAST\_INTERVAL from dba\_hist\_snapshot s  JOIN dba\_hist\_database\_instance i ON (s.dbid = i.dbid and s.instance\_number = i.instance\_number and s.startup\_time = s.startup\_time) group by s.dbid, i.db\_name,i.host\_name,i.platform\_name order by s.DBID;  exit  EOF |

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* Replace #DBID#, #LOWSNAP#,#HIGHSNAP# with values from report above

|  |
| --- |
| oradebug setmypid  oradebug unlimit  oradebug tracefile\_name  alter session set events '942 trace name ERRORSTACK level 3';  begin  dbms\_workload\_repository.drop\_snapshot\_range(low\_snap\_id=>#LOWSNAP#,high\_snap\_id=># HIGHSNAP#,dbid=>#DBID#);  end;  /  exec dbms\_swrf\_internal.unregister\_database(#DBID#); |

### Record <R01-LOG-05> AWR Import Output in the log file

Record <R01-LOG-05> AWR Import Output in the "System Capacity Test - Log" file

### Verify Import

Report available AWRs in the repository and check if your DB is registered

|  |
| --- |
| . oraenv  AWRRPT  sqlplus -s / as sysdba<<EOF  set verify off echo off feed off  set lin 1000  set pagesize 60  column MIN\_SNAP format 99990  column MAX\_SNAP format 99990  column LAST\_INTERVAL format A25  column FIRST\_INTERVAL format A25  column DB\_NAME format A10  column HOST\_NAME format A10 WRAP  column PLATFORM\_NAME format A10 WRAP  select s.DBID, i.db\_name,i.host\_name,substr(i.platform\_name,1,10) platform\_name, min(SNAP\_ID) MIN\_SNAP, max(SNAP\_ID) MAX\_SNAP, min(BEGIN\_INTERVAL\_TIME) FIRST\_INTERVAL, max(BEGIN\_INTERVAL\_TIME) LAST\_INTERVAL from dba\_hist\_snapshot s  JOIN dba\_hist\_database\_instance i ON (s.dbid = i.dbid and s.instance\_number = i.instance\_number and s.startup\_time = s.startup\_time) group by s.dbid, i.db\_name,i.host\_name,i.platform\_name order by s.DBID;  exit  EOF |

### GZIP DMP files

|  |
| --- |
| cd /nfsrwora/12c\_migrate/awrdata nohup gzip \*.dmp & |

### Record information about new test into AWR Capacity Test Inventory

**Important! Update Document!**

Create backup before editing the file!

Backup Name: Capacity Test AWR baselines\_bkup\_YYYYMMDD.xlsx

[https://teamsites.iad.ca.inet/ifc/DBA/Oracle Documents/AWR Repository/Administration/Capacity Test AWR baselines.xlsx](https://teamsites.iad.ca.inet/ifc/DBA/Oracle%20Documents/AWR%20Repository/Administration/Capacity%20Test%20AWR%20baselines.xlsx)

# Information Collection RUN2

## Record <R02-LOG-01> Timestamp of the test

Navigate to R02-LOG-01 section of Log document and paste there timestamps of the test

|  |  |
| --- | --- |
| From | To |
| 2001-02-28 11:00:00 | 2001-02-28 12:00:00 |

## Create Directories to hold diagnostics data

Execute this on All nodes

|  |
| --- |
| mkdir -p /home/oracle/perftest/20010228  mkdir -p /home/oracle/perftest/20010228/$(hostname)  mkdir -p /home/oracle/perftest/20010228/$(hostname)/collectionlogs  mkdir -p /home/oracle/perftest/20010228/$(hostname)/tfa  mkdir -p /home/oracle/perftest/20010228/$(hostname)/db/  mkdir -p /home/oracle/perftest/20010228/$(hostname)/db/ash  mkdir -p /home/oracle/perftest/20010228/$(hostname)/db/awr  cd /home/oracle/perftest/20010228 |

## Immediate Collection

### ASH V$ACTIVE\_SESSION\_HISTORY offline dump

Ref: Analysis of Active Session History (Ash) Online and Offline (Doc ID 243132.1)

ashdump\* scripts and post-load processing of MMNL traces (Doc ID 555303.1)

Run this immediately on all RAC instances after the test.

|  |
| --- |
| cd /home/oracle/perftest/20010228 sqlplus / as sysdba<<EOF>/home/oracle/perftest/20010228/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_times.log  col mintime format A25  col maxtime format A25  col minutes format 9999  SELECT to\_char(MIN(SAMPLE\_TIME),'YYYY-MM-DD HH24:MI:SS') as mintime,  to\_char(Max(SAMPLE\_TIME),'YYYY-MM-DD HH24:MI:SS') as maxtime FROM V\$ACTIVE\_SESSION\_HISTORY;  with diff as (select systimestamp - TIMESTAMP'2001-02-28 11:00:00' as diff from dual)  SELECT extract(day from diff.diff)\*24\*60+extract(hour from diff.diff)\*60+extract(minute from diff.diff) as minutes FROM diff;  exit  EOF  cat /home/oracle/perftest/20010228/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_times.log |

In the output of the script, check for "Minutes" column. You will use that number to collect ASH offline dump.

**Example:**

|  |
| --- |
| SQL> SQL> SQL> SQL> 2  MINTIME MAXTIME  ------------------------- -------------------------  2017-03-26 12:38:40 2017-03-26 21:05:40  SQL> 2  MINUTES  -------  4230 |

**DON'T EDIT TEXT BELOW!**

**Copy it to notepad and change the [**oradebug dump ashdump ####**] with the number of minutes from query above**

|  |
| --- |
| cd /home/oracle/perftest/20010228  sqlplus / as sysdba<<EOF|tee /home/oracle/perftest/20010228/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log  oradebug setmypid  oradebug unlimit  oradebug tracefile\_name  REM number below from minutes column  REM REPLACE NUMBER with number from minutes column from QUERY ABOVE (${ORACLE\_SID}\_ash\_dump\_times.log)  oradebug dump ashdump ####;  exit  EOF  cat /home/oracle/perftest/20010228/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log |

Copy produced file to diagnostic directory

All nodes

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/ash    #Get the filename from ASH collection trace file  cp $(egrep '^SQL>[ ].\*\.trc$' \  /home/oracle/perftest/20010228/$(hostname)/collectionlogs/${ORACLE\_SID}\_ash\_dump\_file.log|sed -e 's/^SQL>[ ]//g') \  $(hostname)\_${ORACLE\_SID}\_ASH\_data.trc  ls -l  #Copy ASH file to network location as well  cp $(hostname)\_${ORACLE\_SID}\_ASH\_data.trc /nfsrwora/12c\_migrate/awrdata/ash/ |

### Record <R02-LOG-02> from OEM ASH Graph on both nodes to Log Document

Navigate to R02-LOG-02 section of Log document and paste there screenshot from Cloud Control ASH Analytics window. Use "Custom" button to specify correct timeframe.

Use at least two dimensions:

* SQLID by Wait Class
* SQL Plan Hash Value by Wait Class
* Do Load Map in Advanced mode

### Extract PerfHUB Report (12c Only)

Use report level ALL

Please enter report level [typical]: all

Run this on all instances of RAC

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr  sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/perfhubrpt.sql |

### Extract Real-Time Active ADDM Report (12c Only)

Change default ADDM range

|  |
| --- |
| vi $ORACLE\_HOME/rdbms/admin/rtaddmrpti.sql  change  define default\_report\_duration = 60;  to  define default\_report\_duration = 1440; |

**Identify ADDM real-time snapshots**

Run this on first instance of RAC only

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/rtaddmrpt.sql |

**Check output**

|  |
| --- |
| DB Id REPORT\_ID TIME trigger\_ceise impact  ---------- --------- -------------------- ------------------------- ----------  2654834759 2893 09/02/2017 16:30:51 High Load 3.49  2654834759 2902 09/02/2017 17:21:23 High Load 3.01 |

**Extract ADDM real-time snapshot**

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr  sqlplus / as sysdba @$ORACLE\_HOME/rdbms/admin/rtaddmrpt.sql |

File should use following naming convention:

[active\_rt]\_ADDM\_[YYYYMMDD]\_[HHMMSS]\_[REPORT ID].html

example:

active\_rt\_ADDM\_20170418\_163051\_2893.html

active\_rt\_ADDM\_20170418\_172123\_2902.html

### Collect ASH info

Review ASH Graph, Real Time ADDM, Perf HUB reports.

identify times when the system was busiest. Use CPU, I/O, Redo generation, unusual events to identify busy time.

Script below will use 10min intervals and produce ASH report for various wait classes for 2 instances (RAC).

**DON'T EDIT TEXT BELOW!**

Copy text into notepad and modify Dates and Times.

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr  export TIMES\_TMP="times.txt"  >$TIMES\_TMP  echo "02/28/01 11:00" >> $TIMES\_TMP  echo "02/28/01 11:10" >> $TIMES\_TMP  echo "02/28/01 11:20" >> $TIMES\_TMP  echo "02/28/01 11:30" >> $TIMES\_TMP  echo "02/28/01 11:40" >> $TIMES\_TMP  echo "02/28/01 11:50" >> $TIMES\_TMP  export WAITCLASS\_TMP="waits.txt"  >$WAITCLASS\_TMP  echo CPU >> $WAITCLASS\_TMP  echo Cluster >> $WAITCLASS\_TMP  echo Application >> $WAITCLASS\_TMP  echo Concurrency >> $WAITCLASS\_TMP  echo Other >> $WAITCLASS\_TMP  echo User\_I\_O >> $WAITCLASS\_TMP  echo System\_I\_O >> $WAITCLASS\_TMP  ash\_collection\_rac()  {  V\_INSTANCE="$1"  if [[ $2 = "ALL" ]]; then  V\_WAIT\_CLASS=""  else  V\_WAIT\_CLASS="$2"  fi  V\_EVENTTIME="$3"  V\_EVENTTIME\_FILE="$4"  echo 1 $V\_INSTANCE  echo 2 $V\_WAIT\_CLASS  echo 3 $V\_EVENTTIME  echo 4 $V\_EVENTTIME\_FILE  sqlplus / as sysdba<<EOF  define wait\_class='$V\_WAIT\_CLASS';  define eventdatetime\_param='$V\_EVENTTIME'  define eventdatetime\_file='$V\_EVENTTIME\_FILE'  define dbid='';  define inst\_num='$V\_INSTANCE';  define report\_type = '';  define slot\_width = '';  define target\_sql\_id = '';  define target\_wait\_class = '&&wait\_class';  define target\_session\_id = '';  define target\_service\_hash = '';  define target\_module\_name = '';  define target\_action\_name = '';  define target\_client\_id = '';  define target\_plsql\_entry = '';  define target\_container = '';  define duration='10';  define begin\_time='&&eventdatetime\_param';  define report\_name = ashrpt\_&&inst\_num.\_&&eventdatetime\_file.\_&&wait\_class..html  @?/rdbms/admin/ashrpti.sql  exit  EOF  }  cat $TIMES\_TMP | while read TIMES\_LINE  do  FILE\_EXT=$(echo $TIMES\_LINE|sed -e 's/[/:]//g' -e 's/[ ]/\_/g')  ash\_collection\_rac 1 "ALL" "$TIMES\_LINE" "$FILE\_EXT"  ash\_collection\_rac 2 "ALL" "$TIMES\_LINE" "$FILE\_EXT"  cat $WAITCLASS\_TMP | while read WAITCLASS\_LINE  do  echo $WAITCLASS\_LINE $TIMES\_LINE $FILE\_EXT  ash\_collection\_rac 1 "$WAITCLASS\_LINE" "$TIMES\_LINE" "$FILE\_EXT"  ash\_collection\_rac 2 "$WAITCLASS\_LINE" "$TIMES\_LINE" "$FILE\_EXT"  :  done  :  done |

## Next Day Collection

### Load ASH Dump file into permanent table on AWRRPT database (OPTIONAL)

Move the files from original servers (all members of RAC cluster)

Run this on AWRRPT node (stha8n083)

|  |
| --- |
| mkdir -p /home/oracle/perftest/20010228  mkdir /home/oracle/perftest/20010228/db/awr  cd /home/oracle/perftest/20010228  rm -f ashdump\*  cp /dbabin/dbatools/diagscript/ashdump/12c/\* .  ls -l /nfsrwora/12c\_migrate/awrdata/ash/\*ASH\_data.trc |

|  |
| --- |
| ASH\_FILE="PUT REQUIRED FILE FILE " |

|  |
| --- |
| ls -1 $ASH\_FILE 2>/dev/null|while read line  do  echo processing file $line  ./ashdump\_loader $line  echo done processing file $line  :  done  ls -l \*.bad  echo Verify if you had any records in the bad file |

### Rename Table to a table to contain timestamp format

|  |
| --- |
| export frm=20010228  sqlplus / as sysdba<<EOF  alter table ashdump.ashdump rename to ashdump${frm};  exit  EOF |

### Collect lfsdiag.sql Information (log sync)

On first node only

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr  sqlplus -s / as sysdba @/dbabin/dbatools/diagscript/lfsdiag.sql |

### Record <R02-LOG-03> Current AWR snapshots

Run script below. Select only information about snapshots and database DBID info. Don't run complete report.

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/db/awr  sqlplus / as sysdba @?/rdbms/admin/awrrpt.sql |

Navigate to R02-LOG-03 section of Log document and paste there AWR snapshots for RUN2 start and finish times.

### Create local AWR report

login to tested database node and run command to generate local AWR reports

You will need to specify 4 parameters:

* DBID
* Database name (if RAC not SID but a database name)
* First Snapshot in the test
* Last snapshot in the test

Source code: <https://github.com/dbarkin/awr_auto_reports>

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #BEGIN\_SNAPID# and #END\_SNAPID# with relevant snapshot IDs from R02-LOG-03 section of Log document.

|  |
| --- |
| unalias cp  cd /home/oracle/perftest/20010228/$(hostname)/db/awr  cp /dbabin/dbatools/diagscript/awr\_auto\_reports/\* .  . generate\_local\_awr.sh  generate\_local\_awr 1111111111 DBSID01 #BEGIN\_SNAPID# #END\_SNAPID# |

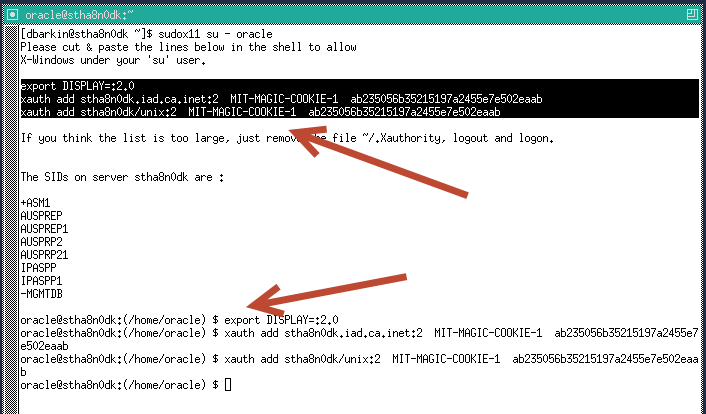
### Start VNC on ALL RAC nodes (required to generate proper graphs using OSWatcher)

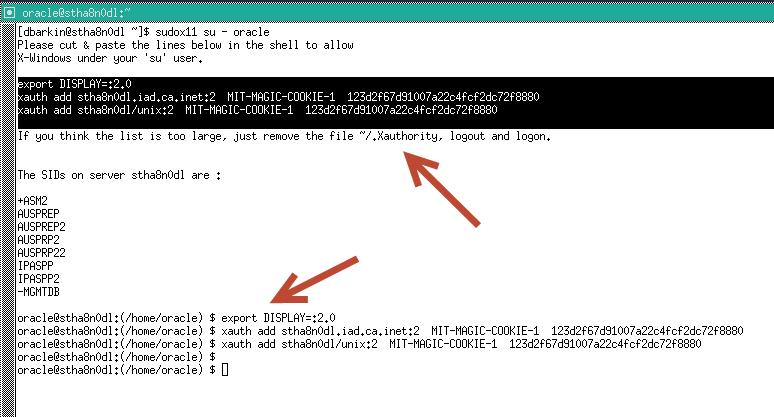
|  |
| --- |
| [dbarkin@stha8n04n ~]$ vncserver :2  VNC(R) Server 5.2.3 (r8648) 64-bit (Jan 28 2015 15:47:49)  Copyright (C) 2002-2015 RealVNC Ltd.  VNC is a registered trademark of RealVNC Ltd in the U.S. and in other  countries.  Protected by UK patent 2481870; US patent 8760366.  See http://www.realvnc.com for information on VNC.  For third party acknowledgements see:  http://www.realvnc.com/products/vnc/documentation/5.2/acknowledgements.txt  Generating private key...done  xauth: creating new authority file /home/dbarkin/.Xauthority  If a desktop environment fails to load for this virtual desktop, please see:  http://www.realvnc.com/kb-345  Running applications in /etc/vnc/xstartup  VNC Server catchphrase: "Spark Joel Kansas. Owner pump sonic."  signature: c7-3b-69-df-64-34-c8-6a  Log file is /home/dbarkin/.vnc/stha8n04n:2.log  New desktop is stha8n04n:2 (10.177.121.74:2)  [dbarkin@stha8n04n ~]$ |

### Open VNC xterminal from your laptop using vncviewer on All RAC Nodes - Required for OSWatcher

|  |  |  |
| --- | --- | --- |
| open VNC xterminal from your laptop using vncviewer on BOTH Nodes   |  |  | | --- | --- | |  |  | |

Make sure you export x11 variables on both nodes





### Run generic TFA diagnostics collection

Run this on all nodes

|  |
| --- |
| export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/20010228/$(hostname)/tfa  >$(hostname)\_tfa\_summary.txt  tfactl summary |tee -a $(hostname)\_tfa\_summary.txt  tfactl events |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp db -from "Feb/28/2001 11:00:00" -to "Feb/28/2001 12:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp os -from "Feb/28/2001 11:00:00" -to "Feb/28/2001 12:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp crs -from "Feb/28/2001 11:00:00" -to "Feb/28/2001 12:00:00" |tee -a $(hostname)\_tfa\_summary.txt  tfactl analyze -comp asm -from "Feb/28/2001 11:00:00" -to "Feb/28/2001 12:00:00" |tee -a $(hostname)\_tfa\_summary.txt  cat $(hostname)\_tfa\_summary.txt  ls -l $(hostname)\_tfa\_summary.txt  #REPLACE Dates with relevant! |

Run this on all nodes

|  |
| --- |
| export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/20010228/$(hostname)/tfa tfactl diagcollect -all -from "Feb/28/2001 11:00:00" -to "Feb/28/2001 12:00:00"|tee /home/oracle/perftest/20010228/$(hostname)/collectionlogs/tfactl\_diagcollect.log  #REPLACE Dates with relevant! |

### Save generic TFA diagnostics collection

Run this on all RAC nodes

|  |
| --- |
| cd /home/oracle/perftest/20010228/$(hostname)/tfa  cp $(ls -1 /su01/app/oracle/tfa/repository/$(ls -1rt /su01/app/oracle/tfa/repository/|tail -1)/\*.zip) .  ls -l \*.zip |

### Run OSW TFA Analysis and GIF graph generation

IMPORTANT!!!

From VNC Session connected to oracle account with `sudox11 su - orace` from xterminal with DISPLAY set and `xauth add` commands

Run this on all RAC nodes

|  |
| --- |
| export DISPLAY=localhost:2  export JAVA\_HOME=/usr/java6/jre  if [ -f /etc/init.d/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.d/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  elif [ -f /etc/init.tfa ]; then  export $(grep TFA\_HOME= /etc/init.tfa)  cd $TFA\_HOME/../..  export TFA\_BASE=`pwd`  $TFA\_HOME/bin/tfactl print status  else  export TFA\_BASE=/su01/app/11.2.0.4/grid/tfa  export TFA\_HOME=/su01/app/11.2.0.4/grid/tfa/$(hostname)/tfa\_home  fi  export PATH=$PATH:$TFA\_HOME/bin  cd /home/oracle/perftest/20010228/$(hostname)/tfa  tfactl oswbb -6 -7 -8 -B Feb 28 11:00:00 2001 -E Feb 28 12:00:00 2017 -S|\  tee /home/oracle/perftest/20010228/$(hostname)/collectionlogs/tfactl\_osw\_gif.log |

### Save OSW TFA Analysis

Archive last OSW TFA collection on all nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/analysis/  cp $(ls -1rt|tail -1) /home/oracle/perftest/20010228/$(hostname)/tfa/$(hostname)\_osw\_analysis.txt  cd /home/oracle/perftest/20010228/$(hostname)/tfa  ls -l $(hostname)\_osw\_analysis.txt |

Save OSW TFA GIF Graphs

Identify gif files produced on both nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/gif/ ls -lrt|tail |tee /home/oracle/perftest/20010228/$(hostname)/collectionlogs/$(hostname)\_osw\_gif\_graphs.log |

Archive information on all nodes

|  |
| --- |
| cd /su01/app/oracle/tfa/repository/suptools/$(hostname)/oswbb/oracle/oswbb/gif/  cp -r $(ls -1rt|tail -1) /home/oracle/perftest/20010228/$(hostname)/tfa/osw\_gif  cd /home/oracle/perftest/20010228/$(hostname)/tfa  ls -l |

### OS Diagnostic Archive

Run this on all RAC nodes

|  |
| --- |
| cd /home/oracle/perftest/20010228  zip -r ../20010228\_OS\_$(hostname).zip \*  zip -sf ../20010228\_OS\_$(hostname).zip |

## AWR Import

### Create Directory Structure on AWRRPT node

Connect to the node which has AWRRPT database (stha8n083 as of 19990131)

|  |
| --- |
| <stha1111> here is the node where the performance test run  mkdir -p /home/oracle/perftest/20010228/stha1111  mkdir -p /home/oracle/perftest/20010228/stha1111/db/awr |

### Create AWR Data directory on Source Database

Execute this on one node

|  |
| --- |
| mkdir -p /nfsrwora/12c\_migrate/awrdata sqlplus / as sysdba <<EOF|tee ${ORACLE\_SID}\_AWR\_datapump\_directoris.log  select \* from global\_name;  create or replace directory AWRDATA as '/nfsrwora/12c\_migrate/awrdata';  create or replace directory AWRLOG as '/nfsrwora/12c\_migrate/awrdata';  set lin 120  column directory\_path format A60  column directory\_name format A30  select directory\_name,directory\_path from dba\_directories where directory\_name like 'AWR%';  exit  EOF |

### Extract AWRs from Source DB

Open "System Capacity - Log" file and Refer to <R02-LOG-03> for Current AWR snapshots

Run this on the node where performance test was performed (<stha1111>)

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #BEGIN\_SNAPID# and #END\_SNAPID# with relevant snapshot IDs from R01-LOG-03 section of Log document.

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee $(date +%Y%m%d%H%M)\_AWR\_load.log  define b\_snap = #BEGIN\_SNAPID#;  define e\_snap = #END\_SNAPID#;  define num\_days = 1;  define db\_name = 'DBSID01';  define dbid = 1111111111;  define directory\_name='AWRDATA';  define begin\_snap = &b\_snap;  define end\_snap = &e\_snap;  column rdate new\_val file\_date  select to\_char(sysdate,'YYMMDD')rdate from dual;  define file\_name=&&db\_name.\_&&begin\_snap.\_&&end\_snap.\_&&file\_date  @$ORACLE\_HOME/rdbms/admin/awrextr.sql  exit  EOF |

### Record <R02-LOG-04> AWR Export Output in the log file

Record AWR Export Output in the "System Capacity Test - Log" file in section <R02-LOG-04>

### Import AWR to AWR Archive database

Connect to the node which has AWRRPT database (stha8n083 as of 19990131)

List required datapump file

|  |
| --- |
| cd /nfsrwora/12c\_migrate/awrdata  ls -lrt \*.dmp|tail |

Export variable with datapump file without .dmp extention

Important: replace NEWDBID!

Important: replace DMPFILE!

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #DMPFILE# with relevant DUMPFILE from "System Capacity Test - Log" file in section <R02-LOG-04>

|  |
| --- |
| . oraenv  AWRRPT  export DMPFILE=#DMPFILE#  # Look for ^^^ space. Remove its  **# Make sure there is no DMP extention^^^^^^**  unset NEWDBID  export NEWDBID=<<<YYMMDDHH24MI>>> of test data Maximum 10 numbers  #Example  export NEWDBID=0102281100 |

Import data

|  |  |
| --- | --- |
| import\_awr(){  export LOG\_FILE=$(date +%Y%m%d%H%M)\_AWR\_load.log  echo IMPORTANT! It will be new DBID $NEWDBID  echo IMPORTANT! Record old DBID and new DBID  sqlplus / as sysdba<<EOF|tee $LOG\_FILE  define directory\_name='AWRDATA';  define schema\_name='AWR\_STAGE';  define file\_name='$DMPFILE';  define default\_tablespace='AWR\_DATA';  define temporary\_tablespace='TEMP';  @$ORACLE\_HOME/rdbms/admin/awrload.sql ${NEWDBID}  exec DBMS\_WORKLOAD\_REPOSITORY.MODIFY\_SNAPSHOT\_SETTINGS( retention=>0, dbid=>${NEWDBID});  exit  EOF  cat $LOG\_FILE  }  if [ -z "$DMPFILE" ]; then  echo "Error! DMPFILE is not set "  else  [[ ! -z "$NEWDBID" ]] && import\_awr || echo "Error! NEWDBID is not set "  fi |  |

**ONLY In Case you need to clean-up failed import**

Identify snapshots and dbid to cleanup

|  |
| --- |
| . oraenv  AWRRPT  sqlplus -s / as sysdba<<EOF  set verify off echo off feed off  set lin 1000  set pagesize 60  column MIN\_SNAP format 99990  column MAX\_SNAP format 99990  column LAST\_INTERVAL format A25  column FIRST\_INTERVAL format A25  column DB\_NAME format A10  column HOST\_NAME format A10 WRAP  column PLATFORM\_NAME format A10 WRAP  select s.DBID, i.db\_name,i.host\_name,substr(i.platform\_name,1,10) platform\_name, min(SNAP\_ID) MIN\_SNAP, max(SNAP\_ID) MAX\_SNAP, min(BEGIN\_INTERVAL\_TIME) FIRST\_INTERVAL, max(BEGIN\_INTERVAL\_TIME) LAST\_INTERVAL from dba\_hist\_snapshot s  JOIN dba\_hist\_database\_instance i ON (s.dbid = i.dbid and s.instance\_number = i.instance\_number and s.startup\_time = s.startup\_time) group by s.dbid, i.db\_name,i.host\_name,i.platform\_name order by s.DBID;  exit  EOF |

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* Replace #DBID#, #LOWSNAP#,#HIGHSNAP# with values from report above

|  |
| --- |
| oradebug setmypid  oradebug unlimit  oradebug tracefile\_name  alter session set events '942 trace name ERRORSTACK level 3';  begin  dbms\_workload\_repository.drop\_snapshot\_range(low\_snap\_id=>#LOWSNAP#,high\_snap\_id=># HIGHSNAP#,dbid=>#DBID#);  end;  /  exec dbms\_swrf\_internal.unregister\_database(#DBID#); |

### Record <R02-LOG-05> AWR Import Output in the log file

Record AWR Import Output in the "System Capacity Test - Log" file in section <R02-LOG-05>.

### Verify Import

Report available AWRs in the repository and check if your DB is registered

|  |
| --- |
| . oraenv  AWRRPT  sqlplus -s / as sysdba<<EOF  set verify off echo off feed off  set lin 1000  set pagesize 60  column MIN\_SNAP format 99990  column MAX\_SNAP format 99990  column LAST\_INTERVAL format A25  column FIRST\_INTERVAL format A25  column DB\_NAME format A10  column HOST\_NAME format A10 WRAP  column PLATFORM\_NAME format A10 WRAP  select s.DBID, i.db\_name,i.host\_name,substr(i.platform\_name,1,10) platform\_name, min(SNAP\_ID) MIN\_SNAP, max(SNAP\_ID) MAX\_SNAP, min(BEGIN\_INTERVAL\_TIME) FIRST\_INTERVAL, max(BEGIN\_INTERVAL\_TIME) LAST\_INTERVAL from dba\_hist\_snapshot s  JOIN dba\_hist\_database\_instance i ON (s.dbid = i.dbid and s.instance\_number = i.instance\_number and s.startup\_time = s.startup\_time) group by s.dbid, i.db\_name,i.host\_name,i.platform\_name order by s.DBID;  exit  EOF |

### GZIP DMP files

|  |
| --- |
| cd /nfsrwora/12c\_migrate/awrdata nohup gzip \*.dmp & |

### Record information about new test into AWR Capacity Test Inventory

**Important! Update Document!**

Create backup before editing the file!

Backup Name: Capacity Test AWR baselines\_bkup\_YYYYMMDD.xlsx

[https://teamsites.iad.ca.inet/ifc/DBA/Oracle Documents/AWR Repository/Administration/Capacity Test AWR baselines.xlsx](https://teamsites.iad.ca.inet/ifc/DBA/Oracle%20Documents/AWR%20Repository/Administration/Capacity%20Test%20AWR%20baselines.xlsx)

# Post-Capacity Activities 2 of 2

## Enable Automation

Ref: [https://teamsites.iad.ca.inet/ifc/DBA/Oracle Documents/Automation/Documentation/Exclude a server from the automation.doc](https://teamsites.iad.ca.inet/ifc/DBA/Oracle%20Documents/Automation/Documentation/Exclude%20a%20server%20from%20the%20automation.doc)

|  |  |  |
| --- | --- | --- |
| Connect on stha24045  cd /oracle\_automation\_nonprod/config  vi odb\_AutomationServerLists.cfg  search for servers participating in capacity testing  Comment out the line for the server to be excluded   |  | | --- | | #PREP stha24541 NOALIAS STANDALONE ORA\_9|ORA\_10|ORA\_11|ORA\_12 |   At the end it should look like this:   |  | | --- | | PREP stha24541 NOALIAS STANDALONE ORA\_9|ORA\_10|ORA\_11|ORA\_12 | |

## Set Audit as per baseline - OPTIONAL (AVOID rebooting DB)

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee ${ORACLE\_SID}\_posttest\_audit.log  show parameter audit  alter system set AUDIT\_TRAIL=OS scope=spfile;  alter system set audit\_syslog\_level=LOCAL3.WARNING scope=spfile;  alter system set statistics\_level = typical;  exit  EOF |

## Set statistics back parameters - OPTIONAL (AVOID rebooting DB)

|  |
| --- |
| sqlplus / as sysdba<<EOF|tee ${ORACLE\_SID}\_posttest\_parameters.log  alter system reset "\_use\_single\_log\_writer" scope=spfile;  exit  EOF |

## Reboot DB ONLY if required - OPTIONAL (AVOID rebooting DB)

|  |
| --- |
| srvctl stop database -d DBSID99  srvctl start database -d DBSID99  srvctl start service -db DBSID99 -service DBSID99\_HTTP2.INTACT.NET  srvctl start service -db DBSID99 -service DBSID99\_HTTP1.INTACT.NET |

# COMPARE 2 RUNS REPORTS

## IDENTIFY AWR Snapshots to Compare (New Run VS Previous baseline run)

* Open Log document:  
  "System Capacity Test - Log <Run1DBSID> <DATE of Run1 YYYYMMDD> <Run2DBSID> <DATE of Run2 YYYYMMDD>"
* Check R01-LOG-03 and R02-LOG-03 Sections for begin and end snapshots.

## Record <R03-LOG-01> in the log file runs to be compared

Connect to AWRRPT database and run query below:

Report available AWRs in the repository and check if your DB is registered

|  |
| --- |
| . oraenv  AWRRPT  sqlplus -s / as sysdba<<EOF  set verify off echo off feed off  set lin 1000  set pagesize 60  column MIN\_SNAP format 99990  column MAX\_SNAP format 99990  column LAST\_INTERVAL format A25  column FIRST\_INTERVAL format A25  column DB\_NAME format A10  column HOST\_NAME format A10 WRAP  column PLATFORM\_NAME format A10 WRAP  select s.DBID, i.db\_name,i.host\_name,substr(i.platform\_name,1,10) platform\_name, min(SNAP\_ID) MIN\_SNAP, max(SNAP\_ID) MAX\_SNAP, min(BEGIN\_INTERVAL\_TIME) FIRST\_INTERVAL, max(BEGIN\_INTERVAL\_TIME) LAST\_INTERVAL from dba\_hist\_snapshot s  JOIN dba\_hist\_database\_instance i ON (s.dbid = i.dbid and s.instance\_number = i.instance\_number and s.startup\_time = s.startup\_time) group by s.dbid, i.db\_name,i.host\_name,i.platform\_name order by s.DBID;  exit  EOF |

**Identify 2 runs to be compared and record them in the "System Capacity Test - Log" file section <R03-LOG-01>**

## Create AWR Difference report (New Run VS Previous baseline run)

DBIDs

|  |  |
| --- | --- |
| 1st run DBID (converted) | 9931010100 |
| 2nd run DBID (converted) | 0102281100 |

Source code <https://github.com/dbarkin/awr_auto_reports>

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace #run1\_snap\_1st#, #run1\_snap\_2nd#, #run2\_snap\_1st#, #run2\_snap\_2nd# with relevant snapshot IDs for each run. Snapshot IDs should contain performance data with similar load.

|  |
| --- |
| . oraenv  AWRRPT  unalias cp  cd /home/oracle/perftest/20010228/stha1111/db/awr  cp /dbabin/dbatools/diagscript/awr\_auto\_reports/\* .  . generate\_compare\_awr.sh  generate\_compare\_awr 9931010100 #run1\_snap\_1st# #run1\_snap\_2nd# 0102281100 #run2\_snap\_1st# #run2\_snap\_2nd#  # ^^^^^^^^^^ ^^^^^^^^^^  # DBID for RUN1 DBID for RUN2 |

## Create TOP SQL report (New Run VS Previous baseline run)

Use AWRRPT database

### Create required view

|  |
| --- |
| . oraenv  AWRRPT  cd /home/oracle/perftest/20010228/stha1111/db/awr  sqlplus / as sysdba<<EOF  @sql\_ranked\_view.sql  exit  EOF |

### Create HTML report to compare SQL performance profile of First test run versus Second test run for busiest snapid

Select busiest snapid for test runs. Try selecting snapid about 1 hour from the start time OR busiest snapids.

dbid\_1st : DBID from first test run

instance\_1st : INSTANCE first test run

snap\_1st : Snapshot ID you are going to use for first test run

dbid\_2nd : DBID from second test run

instance\_2nd : INSTANCE second test run

snap\_2nd : Snapshot ID you are going to use for first test run

|  |  |
| --- | --- |
| dbid\_1st : DBID from first test run | 9931010100 |
| instance\_1st : INSTANCE first test run | 1 |
| snap\_1st : Snapshot ID you are going to use for first test run | 761 <- REPLACE with relevant snapid Use LOG file R03-LOG-01 section for reference |
| dbid\_2nd : DBID from second test run | 0102281100 |
| instance\_2nd : INSTANCE second test run | 1 |
| snap\_2nd : Snapshot ID you are going to use for first test run | 761 <- REPLACE with relevant snapid  Use LOG file R03-LOG-01 section for reference |

**DON'T EDIT TEXT BELOW!**

* Copy text from table below into notepad or similar text editing application.
* In Notepad replace snap\_1st and snap\_2nd with relevant snapshot IDs for each run. Snapshot IDs should contain performance data with similar load.

|  |
| --- |
| cd /home/oracle/perftest/20010228/stha1111/db/awr  sqlplus / as sysdba<<EOF  define dbid\_1st = 9931010100;  define instance\_1st = 1;  define snap\_1st = 761;  define dbid\_2nd = 0102281100;  define instance\_2nd = 1;  define snap\_2nd = 490;  @report\_on\_top\_sqls.sql  exit  EOF |

### Check if report is created

|  |
| --- |
| cd /home/oracle/perftest/20010228/stha1111/db/awr  ls -l \*.html |

### Format reports

* Transfer \*.html to your local computer and open them with excel.
* For each document, select content of the document and
  + change font size to 11
  + change row height to 15
  + format content "Format as Table"
  + save the document as excel format

### Create AWR SQL Report for Each SQL in question (this could be done any time once AWR is populate into AWRRPT)

This will create report for SQL with load profile, complete SQL text, and SQL Plan. Report is useful to verify if SQL has the same plan and load profile during both runs.

Run this on AWRRPT database node - This will create functions to create SQL AWR reports

Table 1 Create functions to run awr reports for SQLs

|  |
| --- |
| . oraenv  AWRRPT  cd /home/oracle/perftest/20010228/stha1111/db/awr  generate\_top\_list() {  sqlplus -s / as sysdba <<EOF  set headsep off heading off term off echo off pagesize 0 linesize 32767 trimspool on FEEDBACK OFF verify off  spool top\_sql\_list.txt  WITH first as (  select \* from sys.SQLS\_RANKED  where DBID='$1' and snap\_id=$2 and INSTANCE\_NUMBER=1)  select  first.sql\_id  from first  where first.rank\_elapsed\_time<11  order by first.rank\_elapsed\_time;  spool off  exit  EOF  }  generate\_awrsql() {  sqlplus / as sysdba<<EOF  define dbid\_a = $1;  define b\_snap = $2;  define e\_snap = $3;  define num\_days = 1;  define db\_name = $4;  define dbid = &dbid\_a;  define begin\_snap = &b\_snap;  define end\_snap = &e\_snap;  define top\_n\_sql = 100;  define inst\_num = 1;  define report\_type = 'html'  define sql\_id = $5  define report\_name = awrsql\_&sql\_id.\_&&dbid.\_&&db\_name.\_&&begin\_snap.\_&&end\_snap..html  @?/rdbms/admin/awrsqrpi.sql  exit  EOF  }  run\_awr\_sql\_report() {  generate\_top\_list $dbid $b\_snapid $e\_snap\_id  cat top\_sql\_list.txt|while read READ\_LINE  do  generate\_awrsql $dbid $b\_snapid $e\_snapid $dbname $READ\_LINE  :  done  } |

**DON'T EDIT TEXT BELOW!**

* Copy text from 2 tables below into notepad or similar text editing application.
* In Notepad replace b\_snapid and e\_snapid with relevant snapshot IDs for each run. Snapshot IDs should contain performance data with similar load.

Table 2 SQL AWR report for top 10 SQL for RUN1

|  |
| --- |
| cd /home/oracle/perftest/20010228/stha1111/db/awr  dbid=9931010100  dbname=DBSID01  b\_snapid=761  e\_snapid=762  run\_awr\_sql\_report |

Table 3 SQL AWR report for top 10 SQL for RUN2

|  |
| --- |
| cd /home/oracle/perftest/20010228/db/awr  dbid=0102281100  dbname=DBSID99  b\_snapid=490  e\_snapid=491  run\_awr\_sql\_report |

These reports could be used to validate SQL load profile, Execution plans.

## Create AWR Diagnostic Compare Archive (New Run VS Previous baseline run)

Run this on AWRRPT database node

|  |
| --- |
| cd /home/oracle/perftest/20010228/stha1111/db/awr  zip -r ../20010228\_AWR\_DBSID01.zip \* |

## Create Diagnostic Archive

Run this on both nodes

|  |
| --- |
| cd /home/oracle/perftest/20010228/stha1111/  zip -r ../20010228\_perf\_$(hostname).zip \*  zip -sf ../20010228\_perf\_$(hostname).zip  ls -l ../\*.zip |

# APPENDIX A

SQL to identify blockes

|  |
| --- |
| with blocked as (  select INST\_ID,SAMPLE\_ID,SAMPLE\_TIME,MODULE,ACTION,CLIENT\_ID,SESSION\_ID,SESSION\_SERIAL#,sql\_id,  BLOCKING\_SESSION ,BLOCKING\_SESSION\_SERIAL# ,BLOCKING\_INST\_ID ,BLOCKING\_HANGCHAIN\_INFO ,CURRENT\_OBJ#,  event,p1text,p1,p2text,p2,p3text,p3 from gv$active\_session\_history where  time\_waited>0  and sample\_time between TIMESTAMP'2017-01-30 19:02:00' and TIMESTAMP'2017-01-30 19:04:00'  and blocking\_session\_status='VALID'  and inst\_id=1 and BLOCKING\_INST\_ID=inst\_id ),  blocker as  (  (  select INST\_ID,SAMPLE\_ID,SAMPLE\_TIME,MODULE,ACTION,CLIENT\_ID,SESSION\_ID,SESSION\_SERIAL#,sql\_id,  BLOCKING\_SESSION ,BLOCKING\_SESSION\_SERIAL# ,BLOCKING\_INST\_ID ,BLOCKING\_HANGCHAIN\_INFO ,CURRENT\_OBJ#,  event,p1text,p1,p2text,p2,p3text,p3 from gv$active\_session\_history where  time\_waited>0  and sample\_time between TIMESTAMP'2017-01-30 19:02:00' and TIMESTAMP'2017-01-30 19:04:00'  and inst\_id=1 and BLOCKING\_INST\_ID=inst\_id )  )  select blocker.sample\_id, blocker.sample\_time,  blocker.MODULE as blocker\_module,blocker.ACTION as blocker\_action,  blocked.MODULE as blocked\_module,blocked.ACTION as blocked\_action,  blocker.sql\_id as blocker\_sql\_id,blocker.CURRENT\_OBJ# blocker\_obj,blocker.event as blocker\_event,  blocked.sql\_id as blocked\_sql\_id,blocked.CURRENT\_OBJ# blocked\_obj,blocked.event as blocked\_event,  blocker.inst\_id, blocker.session\_id blocker\_session\_id,blocker.session\_serial# blocker\_serial#,  blocked.session\_id blocked\_session\_id,blocked.session\_serial# blocked\_serial#,  blocker\_sql.sql\_text as blocker\_sql,  blocked\_sql.sql\_text as blocked\_sql  from blocked  join blocker on blocked.inst\_id=blocker.inst\_id and TO\_CHAR(blocked.sample\_time,'HH24:MI:SS')=TO\_CHAR(blocker.sample\_time,'HH24:MI:SS') and  blocked.BLOCKING\_SESSION=blocker.session\_id and blocked.BLOCKING\_SESSION\_SERIAL#=blocker.session\_serial#  join v$sql blocker\_sql on blocker.sql\_id=blocker\_sql.sql\_id  join v$sql blocked\_sql on blocked.sql\_id=blocked\_sql.sql\_id  order by blocker.sample\_id,blocker.sql\_id,blocker.sql\_id; |

# APPENDIX B

## Log Writer Performance

Reference:

More internals about post-wait or poll-wait mechanism can also be found in Frit's slides (21 to 23): <https://fritshoogland.files.wordpress.com/2015/09/profiling-the-logwriter-and-database-writer.pdf>  
Log file sync from neither I/O nor CPU https://savvinov.com/2014/10/08/log-file-sync-from-neither-io-nor-cpu/   
Tanel Poder, Understanding LGWR, Log File Sync Waits and Commit Performance, <http://files.e2sn.com/slides/Tanel_Poder_log_file_sync.pdf>  
Tuning 'log file sync' wait events: <https://orainternals.wordpress.com/2008/07/07/tuning-log-file-sync-wait-events/>  
Bug 12614085 - Diagnostic enhancement to add new statistics for investigating "log file sync" and "log file parallel write" relationship (Doc ID 12614085.8) ("redo synch time overhead")

gc cr block busy => The consistent read request was delayed, most likely an I/O bottleneck. Blocks are busy in another instance, check for block level contention or hot blocks.

Contention-oriented: The gc current block busy and gc cr block busy events indicate that the remote instance received the block after a remote instance processing delay, in many cases due to a log flush. High concurrency is evidenced by the gc buffer busy event which indicates that the block was pinned or held up by a session on a remote instance. It can also indicate that a session on the same instance has already requested the block.

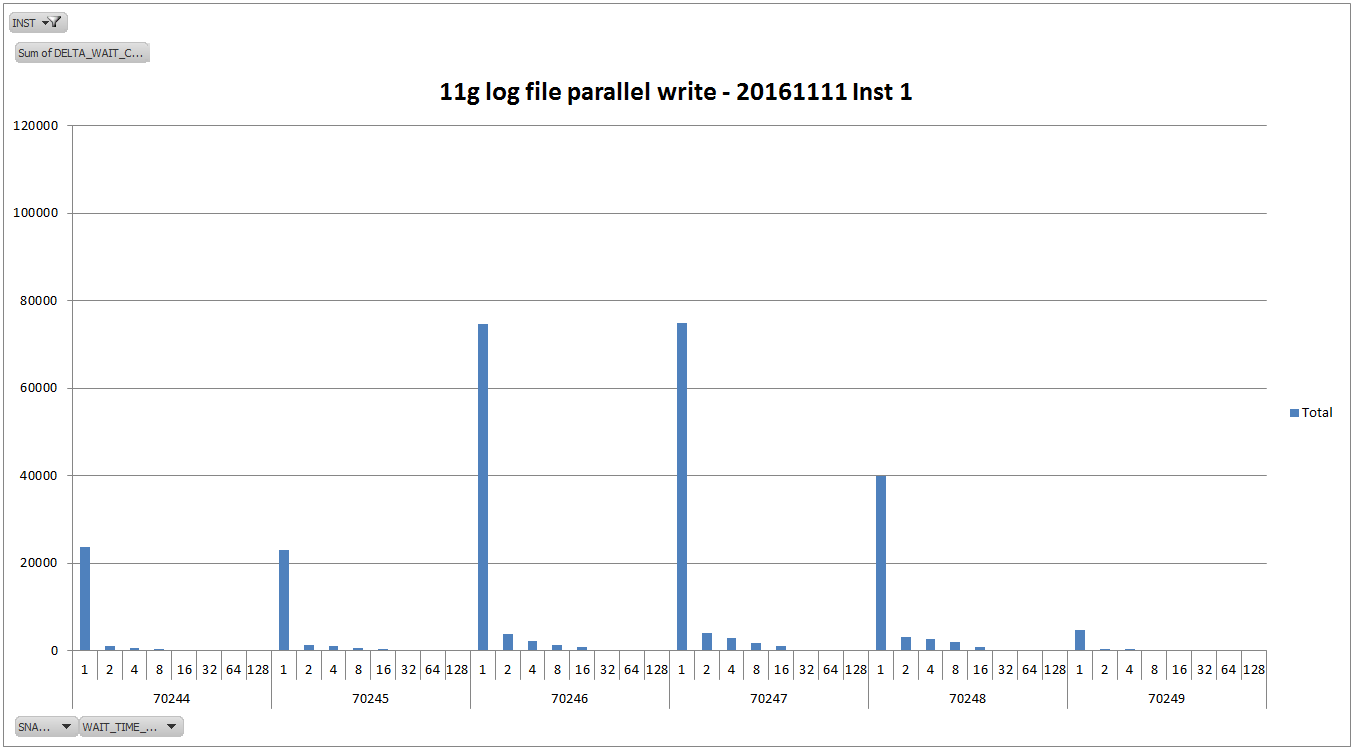
• gc current block busy

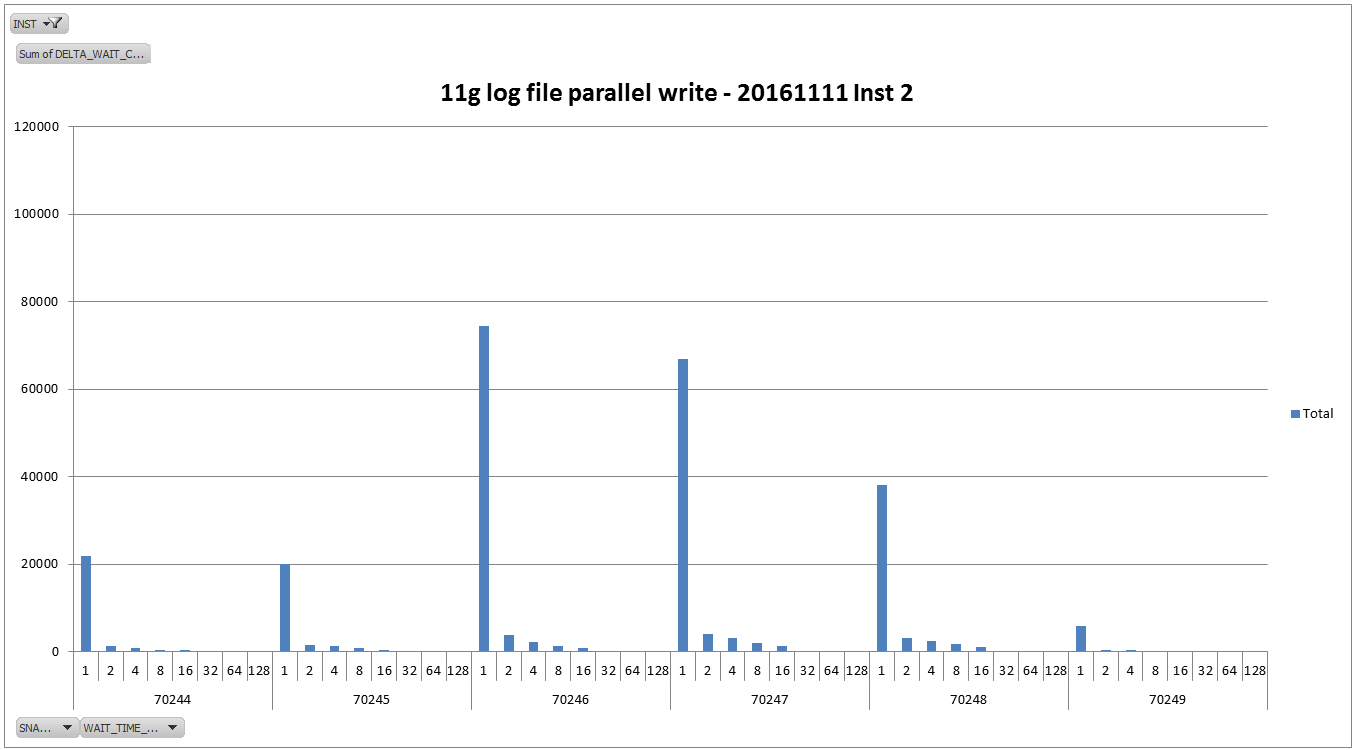
• gc cr block busy

• gc buffer busy

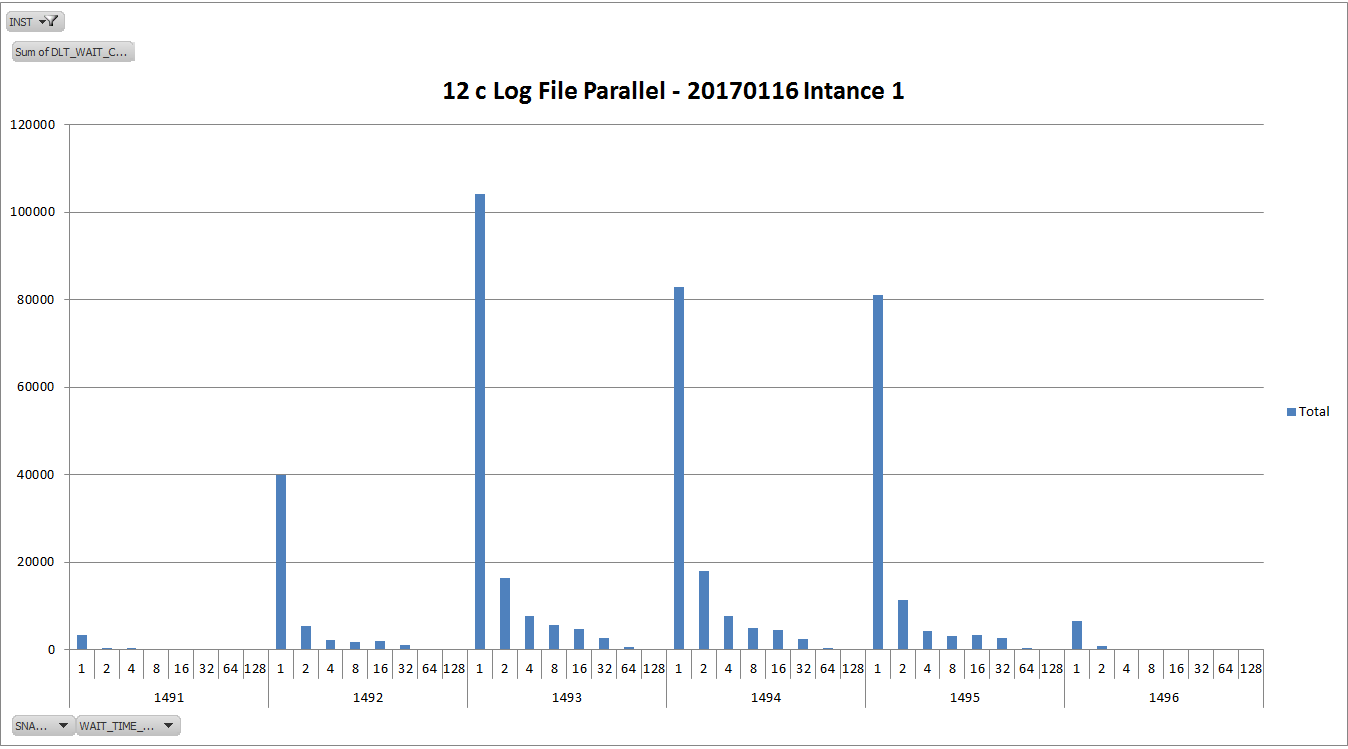
Gcs log flush sync: before sending a current mode block LMS process will request LGWR for a log flush. Until LGWR sends a signal back to LMS process, LMS process will wait on 'gcs log flush' event.

## Log Parallel Write Latency Graph (include excel file C:\Users\dbarkin\Documents\Projects\Savers12c\19990131\log-parallel-latency-histogram\_19990131\_vs\_20161111.xlsx)

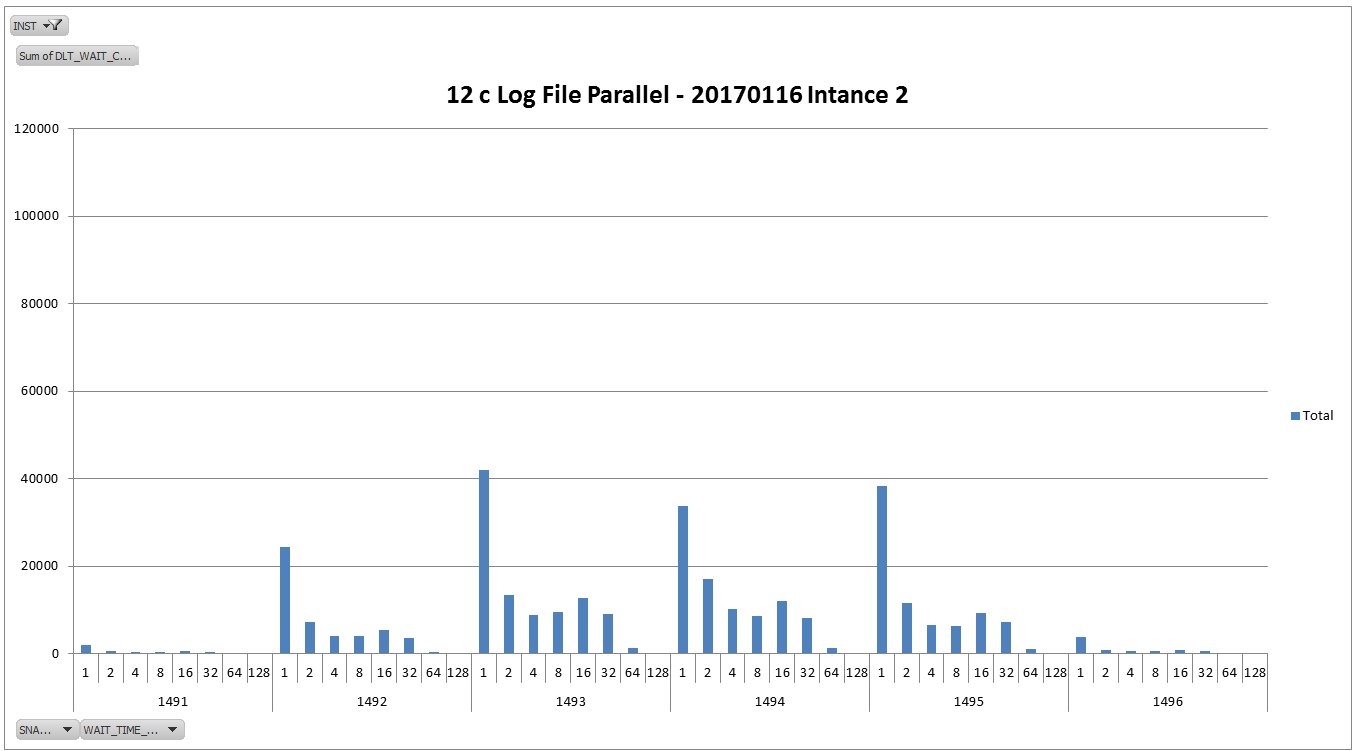
11g Log file parallel write instance 1  


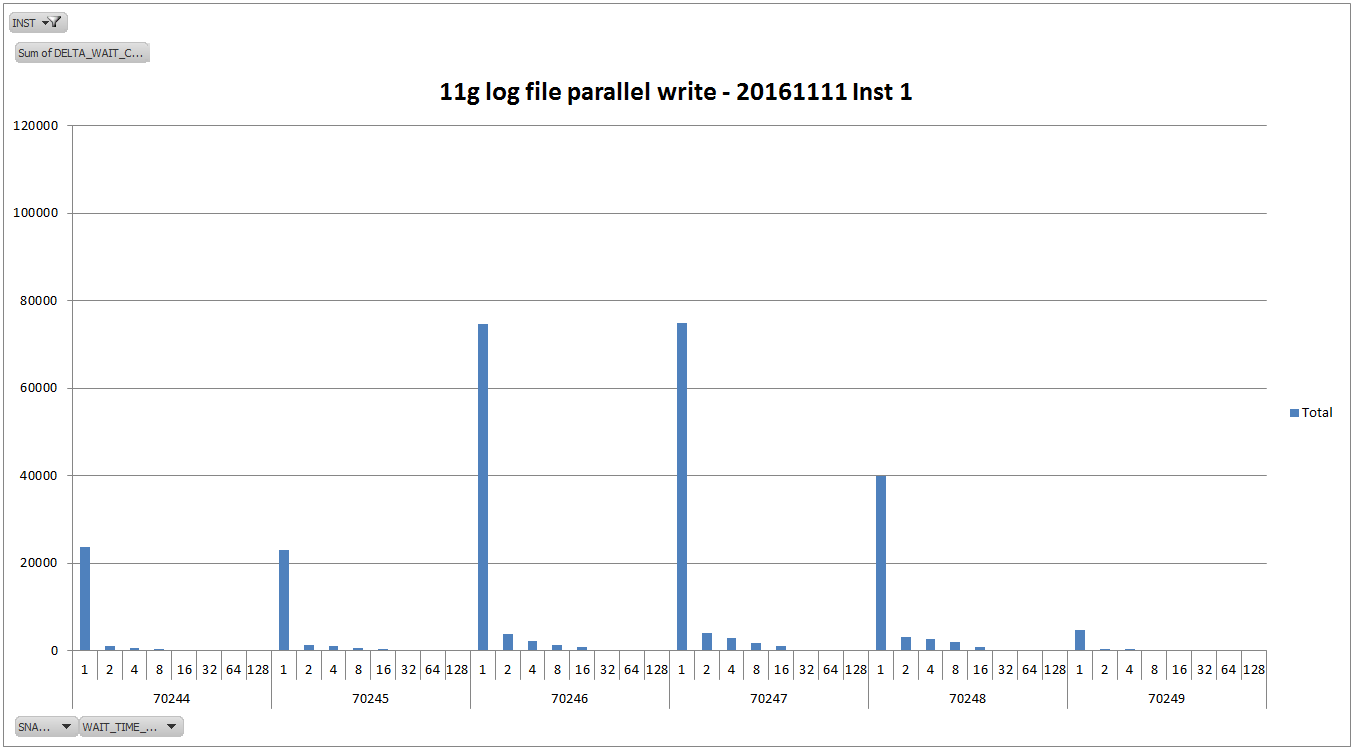


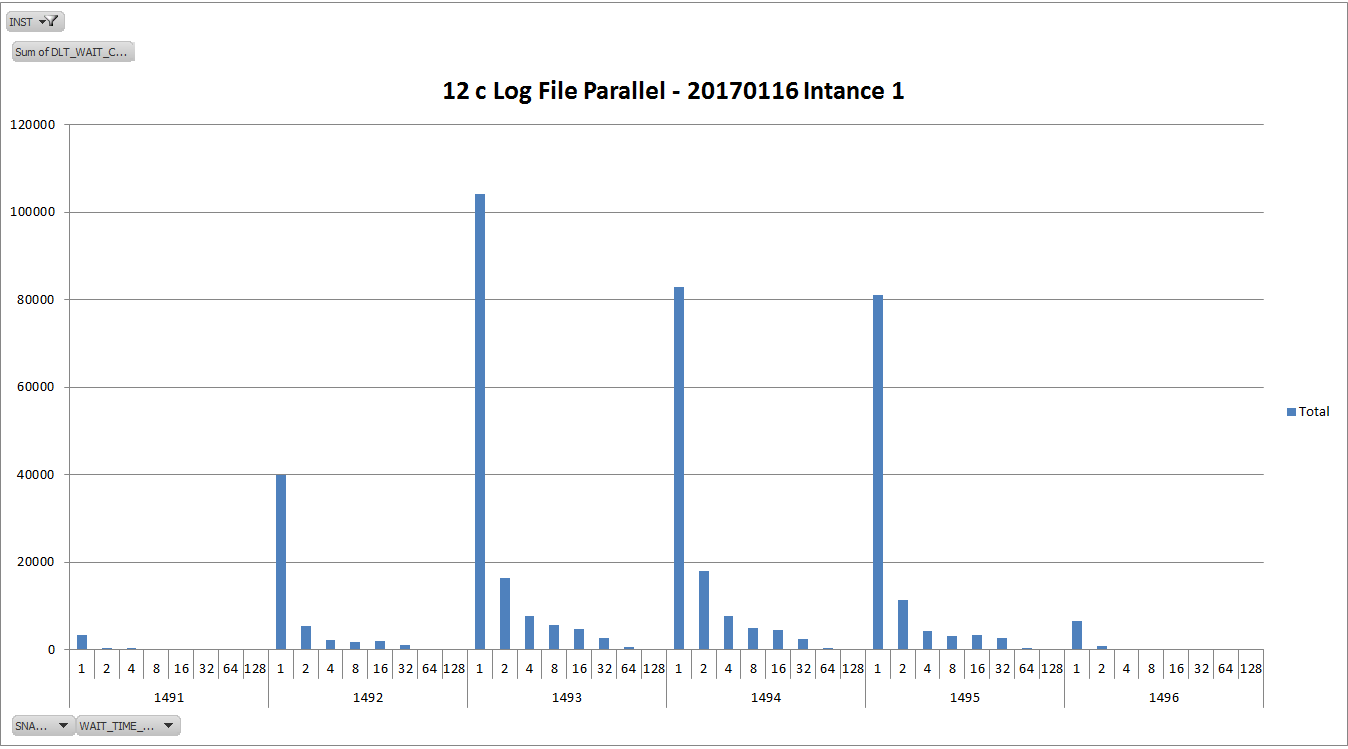
12c Log file parallel write instance 1

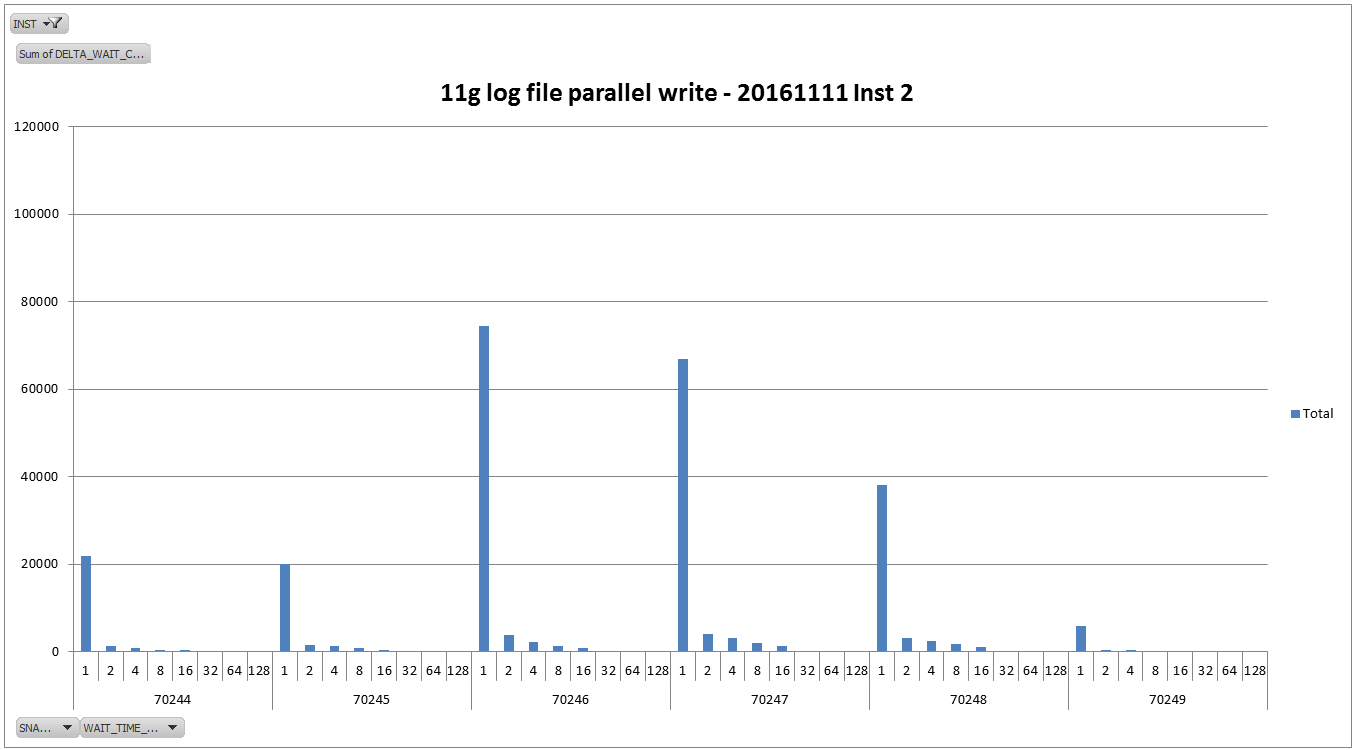


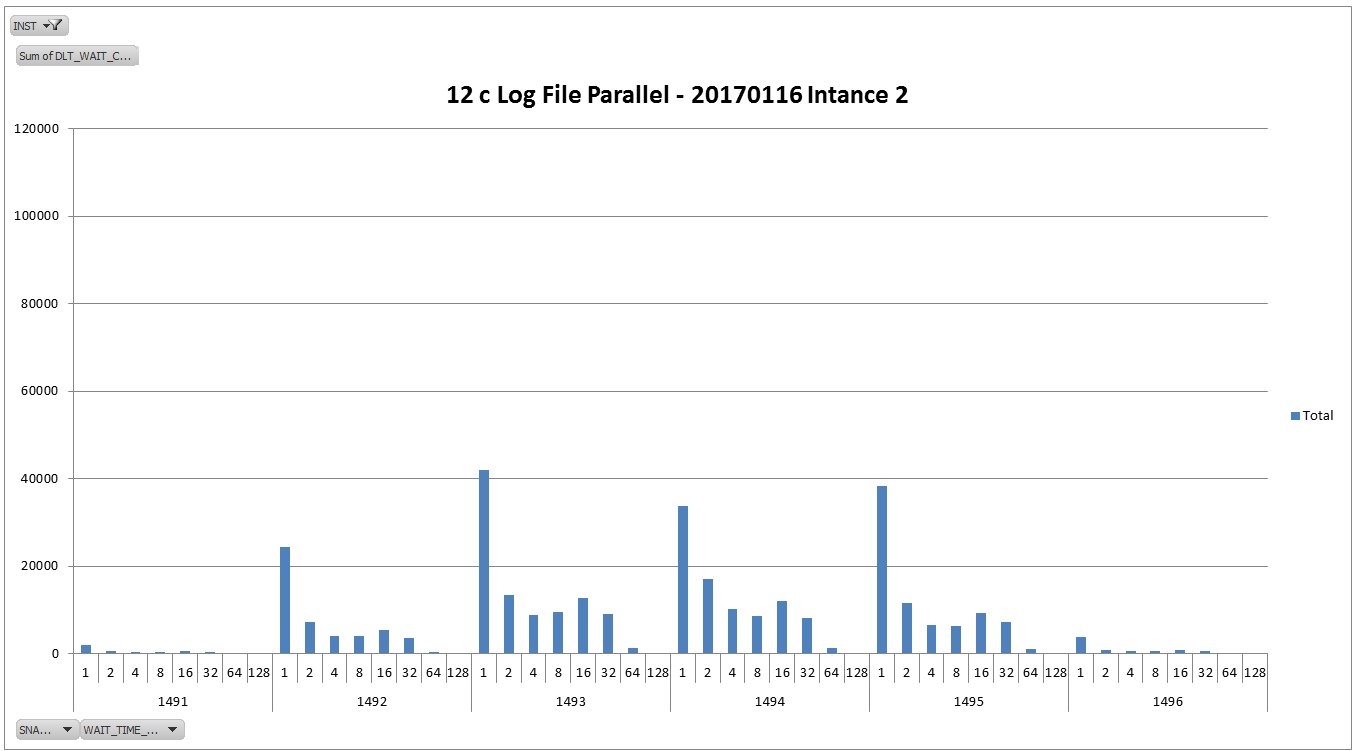
12c Log file parallel write instance 2











### Log Parallel Write Latency Graph - Finding

1. 11g log parallel write is balanced between instances.
2. 12c log parallel write is NOT balanced between instances. Load is higher on instance 1.
3. 12c log parallel write count is much higher than in 11g

Table 4 Percent increase in count of log file parallel write events grouped by latency (ms) for instance 1 (12c over 11g)

|  |  |  |  |
| --- | --- | --- | --- |
| ms | Average Percent Increase of count of log file parallel write events 12c over 11g (for 1 hour period) | 12c (one sample from extract) count of log file parallel write events | 11g (one sample from extract) count of log file parallel write events |
| 1 | 41% | 39935 | 23055 |
| 2 | 311% | 5438 | 1408 |
| 4 | 169% | 2303 | 1069 |
| 8 | 213% | 1882 | 746 |
| 16 | 408% | 2006 | 368 |
| 32 | 1465% | 1167 | 71 |
| 64 | 1475% | 216 | 17 |
| 128 | 489% | 11 | 2 |

### Queries used to retrieve latency information

12c

select snap\_id, inst, name, wait\_time\_milli, wait\_count, prev\_wait\_count,

(CASE WHEN Delta\_wait\_count<0 THEN WAIT\_COUNT -- was reboot just before first snapshots, then we can use current wait\_count as real delta

WHEN Delta\_wait\_count=wait\_count THEN 0 -- It is for first snapshot which has to be disgarded in analysis, required only to get fist counter number

ELSE Delta\_wait\_count

END) Dlt\_wait\_count

from (

select eh.snap\_id, eh.instance\_number inst, en.name, eh.wait\_time\_milli, eh.wait\_count,

LAG(eh.wait\_count, 1, 0) OVER (PARTITION BY eh.instance\_number,eh.wait\_time\_milli ORDER BY eh.snap\_id,eh.instance\_number,eh.wait\_time\_milli) AS prev\_wait\_count,

eh.wait\_count-LAG(eh.wait\_count, 1, 0) OVER (PARTITION BY eh.instance\_number,eh.wait\_time\_milli ORDER BY eh.snap\_id,eh.instance\_number,eh.wait\_time\_milli) AS Delta\_wait\_count -- calculating delta from previous snapshot

from sys.wrh$\_event\_histogram eh, v$event\_name en

where eh.event\_id = en.event\_id and

en.name in 'log file parallel write'

and snap\_id between 1490 and 1496

order by 1,3 desc,2,4

)

order by 1,3 desc,2,4;

11g

select snap\_id, inst, name, wait\_time\_milli, wait\_count, prev\_wait\_count,

(CASE WHEN Delta\_wait\_count<0 THEN WAIT\_COUNT -- was reboot just before first snapshots, then we can use current wait\_count as real delta

WHEN Delta\_wait\_count=wait\_count THEN 0 -- It is for first snapshot which has to be disgarded in analysis, required only to get fist counter number

ELSE Delta\_wait\_count

END) Dlt\_wait\_count

from (

select eh.snap\_id, eh.instance\_number inst, en.name, eh.wait\_time\_milli, eh.wait\_count,

LAG(eh.wait\_count, 1, 0) OVER (PARTITION BY eh.instance\_number,eh.wait\_time\_milli ORDER BY eh.snap\_id,eh.instance\_number,eh.wait\_time\_milli) AS prev\_wait\_count,

eh.wait\_count-LAG(eh.wait\_count, 1, 0) OVER (PARTITION BY eh.instance\_number,eh.wait\_time\_milli ORDER BY eh.snap\_id,eh.instance\_number,eh.wait\_time\_milli) AS Delta\_wait\_count -- calculating delta from previous snapshot

from sys.wrh$\_event\_histogram eh, v$event\_name en

where eh.event\_id = en.event\_id and

en.name in 'log file parallel write'

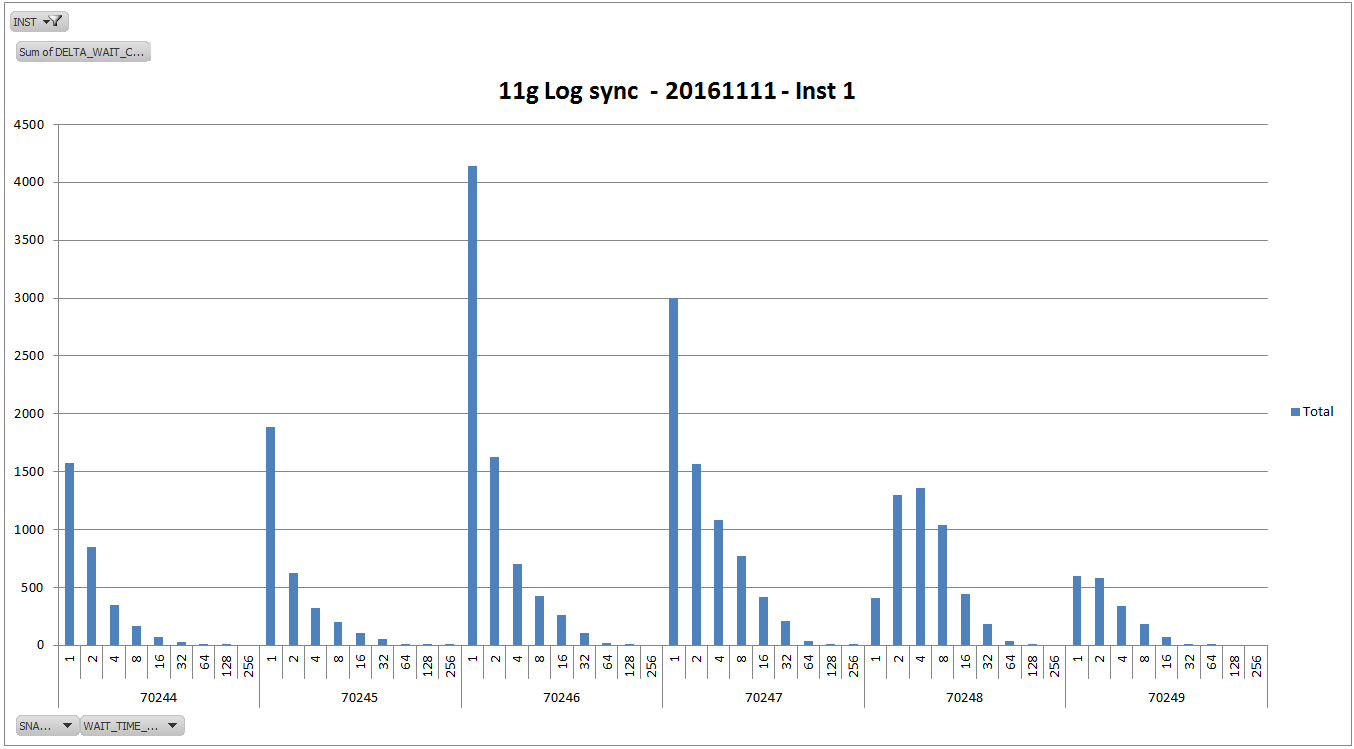
and snap\_id between 70243 and 70249

order by 1,3 desc,2,4

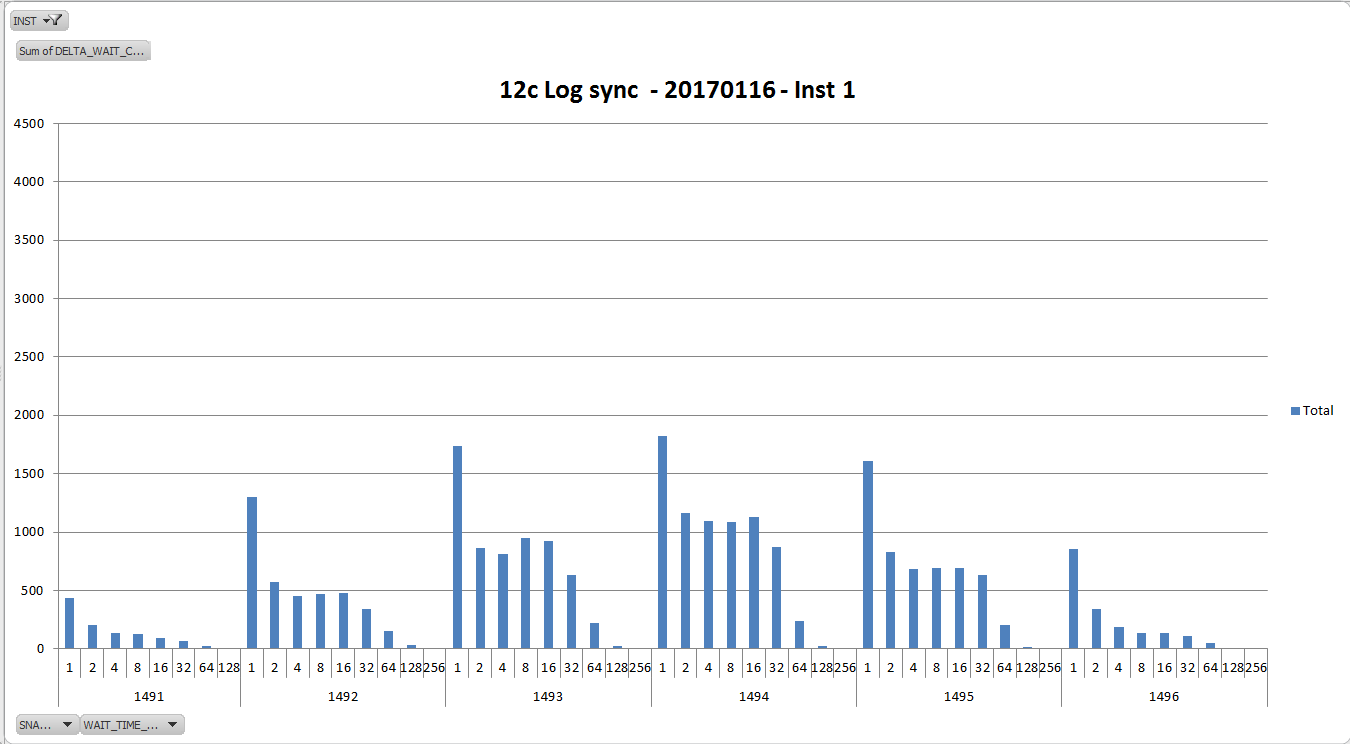
)

order by 1,3 desc,2,4;

11G Log File Sync



12C Log File Sync



# APPENDIX C - Delphix Analytics

|  |
| --- |
| rm -fr ~/1sec  cd /dbabin/delphix/dxtoolkit2  ./dx\_get\_analytics -d dx-eng2-nonprod01 -i 1 -t all -outdir ~/1sec  zip ~/1sec\_$(date +%Y%m%d).zip ~/1sec/\* |

grant necessary privileges and Create trigger

|  |
| --- |
| connect / as sysdba  grant ALTER SESSION to INFA\_ETL;  GRANT administer DATABASE TRIGGER TO "INFA\_ETL";  DROP TRIGGER system.session\_trace;  CREATE OR REPLACE TRIGGER infa\_etl.session\_trace  AFTER LOGON ON DATABASE WHEN (USER = 'INFA\_ETL')  DECLARE  TRC\_NAME VARCHAR2(10);  BEGIN  select to\_char(sysdate,'DDHH24MISS') into TRC\_NAME from dual;  EXECUTE IMMEDIATE 'alter session set timed\_statistics=true';  EXECUTE IMMEDIATE 'alter session set statistics\_level=ALL';  EXECUTE IMMEDIATE 'alter session set max\_dump\_file\_size=unlimited';  EXECUTE IMMEDIATE 'alter session set tracefile\_identifier="INFA'||TRC\_NAME||'"';  DBMS\_APPLICATION\_INFO.SET\_MODULE(module\_name => 'INFA'||TRC\_NAME, action\_name => 'INFA'||TRC\_NAME);  dbms\_application\_info.set\_client\_info('INFA'||TRC\_NAME);  EXECUTE IMMEDIATE 'alter session set "\_px\_trace"="all"';  EXECUTE IMMEDIATE 'alter session set events ''10046 trace name context forever, level 12''';  END;  /  connect infa\_etl/DBSID01\_etl3513  select \* from dual;  exit |

|  |
| --- |
| ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=37aqxrkguq3h1|2rvdg4skmptvv] bind=true, wait=true';  ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=37aqxrkguq3h1|2rvdg4skmptvv] off';  ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=682pg9a3usjt3] bind=true, wait=true';  ALTER SYSTEM SET EVENTS 'sql\_trace [sql: sql\_id=682pg9a3usjt3] off';  ALTER SYSTEM SET EVENTS 'sql\_trace bind=true,wait=true';  ALTER SYSTEM SET EVENTS 'sql\_trace off';  -rw-r----- 1 oracle asmadmin 186 Mar 25 23:42 DBSID01\_p010\_6357316.trm  -rw-r----- 1 oracle asmadmin 7257 Mar 25 23:42 DBSID01\_p010\_6357316.trc  March 25  23:42  EXECUTE IMMEDIATE 'ALTER SESSION SET EVENTS ''sql\_trace bind=true,wait=true''';  Sun Mar 26 00:50:28 EDT 2017  ls -l \*INFA26\*.trc  oracle@stha1111:DBSID01(/su01/app/oracle/diag/rdbms/DBSID01/DBSID01/trace) $ grep \_px\_trace \*.trc  DBSID01\_ora\_5701968.trc: EXECUTE IMMEDIATE 'alter session set "\_px\_trace"="all"';  DBSID01\_ora\_5701968.trc: EXECUTE IMMEDIATE 'alter session set "\_px\_trace"="all"';  oracle@stha1111:DBSID01(/su01/app/oracle/diag/rdbms/DBSID01/DBSID01/trace) $ |

# APPENDIX D - Identify tracefiles with sqlid

Identify SQLIDs using AWR or ASH reports

Using ashrpt\_1\_032717\_0200\_.html

### Top SQL with Top Events

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SQL ID | Planhash | Sampled # of Executions | % Activity | Event | % Event | Top Row Source | % RwSrc | SQL Text |
| [d54hsgzkk2rxn](file:///C:/Users/dbarkin/Documents/ashrpt_1_032717_0200_.html#d54hsgzkk2rxn) | 2840532860 | 1 | 98.37 | CPU + Wait for CPU | 87.30 | TABLE ACCESS - FULL | 86.43 | INSERT INTO POL\_A\_VEHICLES\_RW\_... |
| d54hsgzkk2rxn | 2840532860 | 1 | 98.37133550488599348534201954397394136808 | direct path read | 11.07 | TABLE ACCESS - FULL | 11.07 |  |

|  |  |
| --- | --- |
| d54hsgzkk2rxn | INSERT INTO POL\_A\_VEHICLES\_RW\_CHNG\_ENT SELECT RV.COMPANY\_CD, RV.POLICY\_PROVINCE\_CD, RV.AGREEMENT\_NBR, RV.RISK\_SEQ\_NBR, RV.RECORD\_ID, RV.STAGING\_POLICY\_ID, RV.POL\_A\_VEHICLE\_ID, RV.PREV\_ENTRY\_STAGING\_POLICY\_ID, RV.PREV\_ENTRY\_POL\_VEH\_ID, CV.ENTRY\_DT, RV.POLICY\_INCEPTION\_DT, CV.CHANGE\_EFFECTIVE\_DT, RV.FILE\_DT, 'NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY' AS COLUMN\_NM, CASE WHEN 'NUMBER' = 'NUMBER' THEN TO\_CHAR(PV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY) WHEN 'NUMBER' = 'DATE' THEN TO\_CHAR(PV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY, 'YYYY-MM-DD') ELSE TO\_CHAR(PV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY) END AS PREV\_VALUE, CASE WHEN 'NUMBER' = 'NUMBER' THEN TO\_CHAR(CV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY) WHEN 'NUMBER' = 'DATE' THEN TO\_CHAR(CV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY, 'YYYY-MM-DD') ELSE TO\_CHAR(CV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY) END AS CUR\_VALUE, 414060.000, SYSDATE, 'INFA\_ETL', SYSDATE, 'INFA\_ETL' FROM REL\_PREV\_POL\_VEHICLES RV JOIN POL\_A\_VEHICLES CV ON RV.POL\_A\_VEHICLE\_ID = CV.POL\_A\_VEHICLE\_ID JOIN WRK\_POL\_CHG\_STAGING\_POLICY\_ID W RK ON WRK.STAGING\_POLICY\_ID= RV.STAGING\_POLICY\_ID JOIN POL\_A\_VEHICLES PV ON RV.PREV\_ENTRY\_POL\_VEH\_ID = PV.POL\_A\_VEHICLE\_ID WHERE CASE WHEN 'NUMBER' = 'DATE' THEN NVL (TO\_CHAR(CV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY, 'YYYY-MM-DD'), '?') ELSE NVL (TO\_CHAR(CV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY), '?') END <> CASE WHEN 'NUMBER' = 'DATE' THEN NVL (TO\_CHAR(PV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY, 'YYYY-MM-DD'), '?') ELSE NVL (TO\_CHAR(PV.NBR\_CLM\_5YR\_PR\_NR\_SP\_SA\_QTY), '?') END |

|  |
| --- |
| cd /su01/app/oracle/diag/rdbms/DBSID01/DBSID01/trace  grep d54hsgzkk2rxn \*.trc|awk -F ":" '{print $1}'|uniq -u |

|  |
| --- |
| oracle@stha1111:DBSID01(/su01/app/oracle/diag/rdbms/DBSID01/DBSID01/trace) $ grep d54hsgzkk2rxn \*.trc|awk -F ":" '{print $1}'|uniq -u  DBSID01\_ora\_12648714.trc  DBSID01\_p000\_25296912.trc  DBSID01\_p001\_22544438.trc  DBSID01\_p002\_8061280.trc  DBSID01\_p003\_15401016.trc  DBSID01\_p004\_23592972.trc  DBSID01\_p005\_21758234.trc  DBSID01\_p006\_5243002.trc  DBSID01\_p007\_18612278.trc  DBSID01\_p008\_24445076.trc  DBSID01\_p009\_24576156.trc  oracle@stha1111:DBSID01(/su01/app/oracle/diag/rdbms/DBSID01/DBSID01/trace) $ |

trcsess [output=<output file name >]

|  |
| --- |
| ls -1rt \*INFA\*.trc  DBSID01\_p007\_17760494\_INFA28153811.trc  DBSID01\_p006\_10616990\_INFA28153811.trc  DBSID01\_p005\_19398948\_INFA28153811.trc  DBSID01\_p004\_24444942\_INFA28153811.trc  DBSID01\_p003\_25559152\_INFA28153811.trc  DBSID01\_p002\_12452148\_INFA28153811.trc  DBSID01\_p001\_19464266\_INFA28153811.trc  DBSID01\_p000\_12321124\_INFA28153811.trc  DBSID01\_ora\_12189840\_INFA28153811.trc  vi control.txt |

|  |
| --- |
| cd /su02/dbatools  unzip -o /dbabin/dbatools/trca/trca\_11g.zip -d /su02/dbatools  cd trca/install  sqlplus / as sysdba  START tacreate.sql  Secret#1234  USERS  TEMP  INFA\_ETL  exit |

|  |
| --- |
| cd /su02/dbatools  cd trca/run  sqlplus / as sysdba<<EOF  grant TRCA\_USER\_ROLE to INFA\_ETL;  connect infa\_etl/DBSID01\_etl3513  START trcanlzr.sql control.txt  exit  EOF |

# APPENDIX E Delphix Information collection

Run this on the node where dxtoolkit is installed (it could be run from oracle@stha8n083)

|  |
| --- |
| mkdir -p ~/19990131\_delphix  ./dx\_get\_analytics -d dx-eng2-nonprod01 -i 1 -t all -st "1999-01-31 02:00:00" -et "2017-03-27 06:00:00" -outdir ~/19990131\_delphix  cd ~/19990131\_delphix/  zip /home/oracle/perftest/19990131/delphix.zip \* |