|  |
| --- |
|  |
|  | **Check SGA and PGA Usage statistics**  select sn.INSTANCE\_NUMBER, sga.allo sga, pga.allo pga,(sga.allo+pga.allo) tot,trunc(SN.END\_INTERVAL\_TIME,'mi') time  from  (select snap\_id,INSTANCE\_NUMBER,round(sum(bytes)/1024/1024/1024,3) allo  from DBA\_HIST\_SGASTAT  group by snap\_id,INSTANCE\_NUMBER) sga  ,(select snap\_id,INSTANCE\_NUMBER,round(sum(value)/1024/1024/1024,3) allo  from DBA\_HIST\_PGASTAT where name = 'total PGA allocated'  group by snap\_id,INSTANCE\_NUMBER) pga  , dba\_hist\_snapshot sn  where sn.snap\_id=sga.snap\_id  and sn.INSTANCE\_NUMBER=sga.INSTANCE\_NUMBER  and sn.snap\_id=pga.snap\_id  and sn.INSTANCE\_NUMBER=pga.INSTANCE\_NUMBER  order by sn.snap\_id desc, sn.INSTANCE\_NUMBER  ;  **OR**  select round(used.bytes /1024/1024 ,2) used\_mb  , round(free.bytes /1024/1024 ,2) free\_mb  , round(tot.bytes /1024/1024 ,2) total\_mb  from (select sum(bytes) bytes  from v$sgastat  where name != 'free memory') used  , (select sum(bytes) bytes  from v$sgastat  where name = 'free memory') free  , (select sum(bytes) bytes  from v$sgastat) tot  /  **Find out how much memory each session is using**  COLUMN sid FORMAT 999 HEADING 'SID' |
|  | COLUMN oracle\_username FORMAT a12 HEADING 'Oracle User' JUSTIFY right |
|  | COLUMN os\_username FORMAT a9 HEADING 'O/S User' JUSTIFY right |
|  | COLUMN session\_program FORMAT a18 HEADING 'Session Program' TRUNC |
|  | COLUMN session\_module FORMAT a18 HEADING 'Session module' TRUNC |
|  | COLUMN session\_action FORMAT a18 HEADING 'Session action' TRUNC |
|  | COLUMN session\_machine FORMAT a8 HEADING 'Machine' JUSTIFY right TRUNC |
|  | COLUMN session\_pga\_memory FORMAT 9,999,999,999 HEADING 'PGA Memory' |
|  | COLUMN session\_pga\_memory\_max FORMAT 9,999,999,999 HEADING 'PGA Memory Max' |
|  | COLUMN session\_uga\_memory FORMAT 9,999,999,999 HEADING 'UGA Memory' |
|  | COLUMN session\_uga\_memory\_max FORMAT 9,999,999,999 HEADING 'UGA Memory MAX' |
|  | COLUMN session\_total\_memory FORMAT 9,999,999,999 HEADING 'Total Memory' |
|  |  |
|  | select sid,oracle\_username,os\_username,session\_program,session\_module,session\_action, |
|  | session\_pga\_memory,session\_pga\_memory\_max,session\_uga\_memory,session\_uga\_memory\_max,session\_pga\_memory+session\_uga\_memory session\_total\_memory from ( |
|  | SELECT |
|  | s.sid sid |
|  | , lpad(s.username,12) oracle\_username |
|  | , lpad(s.osuser,9) os\_username |
|  | , s.program session\_program |
|  | , s.module session\_module |
|  | , s.action session\_action |
|  | , lpad(s.machine,8) session\_machine |
|  | , (select ss.value from v$sesstat ss, v$statname sn |
|  | where ss.sid = s.sid and |
|  | sn.statistic# = ss.statistic# and |
|  | sn.name = 'session pga memory') session\_pga\_memory |
|  | , (select ss.value from v$sesstat ss, v$statname sn |
|  | where ss.sid = s.sid and |
|  | sn.statistic# = ss.statistic# and |
|  | sn.name = 'session pga memory max') session\_pga\_memory\_max |
|  | , (select ss.value from v$sesstat ss, v$statname sn |
|  | where ss.sid = s.sid and |
|  | sn.statistic# = ss.statistic# and |
|  | sn.name = 'session uga memory') session\_uga\_memory |
|  | , (select ss.value from v$sesstat ss, v$statname sn |
|  | where ss.sid = s.sid and |
|  | sn.statistic# = ss.statistic# and |
|  | sn.name = 'session uga memory max') as session\_uga\_memory\_max |
|  | FROM |
|  | v$session s ) |
|  | ORDER BY session\_total\_memory DESC; |

|  |
| --- |
|  |
|  | **- Check what a specific session is doing:**  select |
|  | p.SPID UnixProcess ,s.SID,s.serial#,s.USERNAME,s.COMMAND,s.MACHINE,s.SQL\_ADDRESS,s.SQL\_HASH\_VALUE |
|  | ,s.program, status,sql\_text,COMMAND\_TYPE |
|  | from gv$session s,gv$process p, gv$sqlarea sa |
|  | where p.ADDR = s.PADDR and s.inst\_id=p.inst\_id |
|  | and sa.ADDRESS = s.SQL\_ADDRESS and s.inst\_id=sa.inst\_id |
|  | and s.sid=1722; |

**Get TEMP Usage**

**Check Usage first**

select b.Total\_MB,

       b.Total\_MB - round(a.used\_blocks\*8/1024) Current\_Free\_MB,

       round(used\_blocks\*8/1024)                Current\_Used\_MB,

      round(max\_used\_blocks\*8/1024)             Max\_used\_MB

from v$sort\_segment a,

 (select round(sum(bytes)/1024/1024) Total\_MB from dba\_temp\_files ) b;

**Then, check the sessions that use temp tablespace:**

col hash\_value for a40

col tablespace for a10

col username for a15

set linesize 132 pagesize 1000

SELECT s.sid, s.username, u.tablespace, s.sql\_hash\_value||'/'||u.sqlhash hash\_value, u.segtype, u.contents, u.blocks

FROM v$session s, v$tempseg\_usage u

WHERE s.saddr=u.session\_addr

order by u.blocks;

**Check SQL\_ID of that SQL**

select sql\_id from v$session where sid=872;

**Check SQL\_TEXT of that sql\_id.**

select sql\_text from v$sql where sql\_id='';

**Get SID from PID’s**

col sid format 999999

col username format a20

col osuser format a15

select b.spid,a.sid, a.serial#,a.username, a.osuser

from v$session a, v$process b

where a.paddr= b.addr

and b.spid=24759

order by b.spid;

**Current SQL's**

select s.sid,s.schemaname,s.osuser,a.sql\_text

from   v$session s,v$sqlarea a,v$process p

where  s.sql\_hash\_value = a.hash\_value (+)

and    s.sql\_address    = a.address (+)

and    s.paddr          = p.addr;

**For a session**

select s.sid,s.schemaname,s.osuser,a.sql\_text,a.sql\_id

from   v$session s,v$sqlarea a,v$process p

where  s.sql\_hash\_value = a.hash\_value (+)

and    s.sql\_address    = a.address (+)

and    s.paddr          = p.addr and s.sid=&sid

select to\_char(sysdate,'DD-MON HH24:MI:SS') Collection\_datestamp, x.last\_call\_et, x.status, x.sql\_id,

y.sql\_text,y.elspased\_time,y.cpu\_time

from v$session x, v$sqlarea y

where x.sql\_address = y.address

and x.sql\_hash\_value = y.hash\_value

and x.sid = &session\_id

**High Water mark**

SELECT upper(resource\_name) as resource\_name,current\_utilization,max\_utilization,initial\_allocation

FROM v$resource\_limit WHERE resource\_name in ('processes', 'sessions');

select \* from V$license;

**SQL TUNING TASK**

 CREATE or REPLACE PROCEDURE create\_tuning\_task IS

  tuning\_task VARCHAR2(30);

  sqltext     CLOB;

  BEGIN

  sqltext :=   'select  \* from dual';

   tuning\_task := DBMS\_SQLTUNE.CREATE\_TUNING\_TASK(

               sql\_text    => sqltext,

               user\_name   => 'SYS',

               scope       => 'COMPREHENSIVE',

               time\_limit  => 30,

               task\_name   => 'TEST',

               description => 'Tuning effort for counting customer orders');

   END create\_tuning\_task;

   /

SQL > BEGIN

           DBMS\_SQLTUNE.EXECUTE\_TUNING\_TASK( task\_name => 'TEST' );

          END;

SQL > SELECT DBMS\_SQLTUNE.REPORT\_TUNING\_TASK( 'TEST') FROM DUAL;

**Archives Per Day**

   SELECT

   TO\_CHAR(completion\_time,'YYYY-MM-DD') completion\_date,

   round (SUM(block\_size\*(blocks+1)) / 1024 / 1024 / 1024 , 2) GB\_USED\_PER\_DAY

   FROM v$archived\_log

   WHERE TRUNC(completion\_time) BETWEEN

   TRUNC(SYSDATE-30) AND TRUNC(SYSDATE)

   and to\_char(completion\_time,'HH24:MI:SS') >'07:00:00' and

   to\_char(completion\_time,'HH24:MI:SS') < '08:00:00'

   GROUP BY TO\_CHAR(completion\_time,'YYYY-MM-DD')

   order by 1 desc;

**High Water Mark of a Data file**

select file\_name, hwm\*8192/1024/1024, (blocks\*8192)/1024/1024 total\_blocks, ((blocks-hwm+1)\*8192)/1024/1024 shrinkage\_possible

from dba\_data\_files a,

( select file\_id, max(block\_id+blocks) hwm

   from dba\_extents

   group by file\_id ) b

  where a.file\_id = b.file\_id

**Explain Plan**

SQL> EXPLAIN PLAN set statement\_id = 'S1' into plan\_table

  2  FOR

SQL> SELECT \* FROM table(DBMS\_XPLAN.DISPLAY('plan\_table', 'S1','ALL')) ;

@$ORACLE\_HOME/rdbms/admin/awrsqrpt.sql ::   Provide Sql id

-- Shared Pool

            select \* from table(dbms\_xplan.display\_cursor('SQL\_ID',null,'ALL'));

If we add the GATHER\_PLAN\_STATISTICS hint to our simple SQL statement we should be able to see the actual cardinality of each operation at execution time alongside the Optimizer estimates for each cardinality in the plan

SELECT /\*+ GATHER\_PLAN\_STATISTICS \*/ p.prod\_name, SUM(s.quantity\_sold)

            -- AWR

            select \* from table(dbms\_xplan.display\_awr('SQL\_ID',null,null,'ALL'));

            select \* from table(dbms\_xplan.display\_awr('SQL\_ID',null,DBID,'ALL'));

Set Session Trace

EXEC DBMS\_System.Set\_Sql\_Trace\_In\_Session(sid, serial#, true );

EXEC DBMS\_System.Set\_Ev(sid, serial#, event, level, name); EXEC DBMS\_System.Set\_Ev(31, 97, 10046, 4, '');

EXEC DBMS\_System.Set\_Ev(31, 97, 10046, 4, '');

EXEC DBMS\_MONITOR.session\_trace\_enable(session\_id =>1234, serial\_num=>1234, waits=>TRUE, binds=>FALSE);

SQL> -- set the OS PID

SQL> oradebug setospid 864

Windows thread id: 864, image: ORACLE.EXE

SQL> -- or set the Oracle pid

SQL> oradebug setorapid 13

Windows thread id: 864, image: ORACLE.EXE

SQL> -- set the trace file size to unlimitd

SQL> oradebug unlimit

Statement processed.

SQL> -- now turn on trace for SCOTT

SQL> oradebug event 10046 trace name context forever, level 12

Statement processed.

SQL> -- run some queries in another session and then turn trace off

SQL> oradebug event 10046 trace name context off