**ASH and AWR Performance Tuning Scripts**

Listed below are some SQL queries which I find particularly useful for performance tuning. These are based on the Active Session History V$ View to get a current perspective of performance and the DBA\_HIST\_\* AWR history tables for obtaining performance data pertaining to a period of time in the past.

I would like to add that these queries have been customised by me based on SQL scripts obtained from colleagues and peers. So if I am infringing any copyright material let me know and I shall remove the same. Also If anyone has any similar useful scripts to contribute for use by the community do send it to me and I shall include the same on this page

**Top Recent Wait Events**

col EVENT format a60

select \* from (

select active\_session\_history.event,

sum(active\_session\_history.wait\_time +

active\_session\_history.time\_waited) ttl\_wait\_time

from v$active\_session\_history active\_session\_history

where active\_session\_history.event is not null

group by active\_session\_history.event

order by 2 desc)

where rownum < 6

/

**Top Wait Events Since Instance Startup**

col event format a60

select event, total\_waits, time\_waited

from v$system\_event e, v$event\_name n

where n.event\_id = e.event\_id

and n.wait\_class !='Idle'

and n.wait\_class = (select wait\_class from v$session\_wait\_class

where wait\_class !='Idle'

group by wait\_class having

sum(time\_waited) = (select max(sum(time\_waited)) from v$session\_wait\_class

where wait\_class !='Idle'

group by (wait\_class)))

order by 3;

**List Of Users Currently Waiting**

col username format a12

col sid format 9999

col state format a15

col event format a50

col wait\_time format 99999999

set pagesize 100

set linesize 120

select s.sid, s.username, se.event, se.state, se.wait\_time

from v$session s, v$session\_wait se

where s.sid=se.sid

and se.event not like 'SQL\*Net%'

and se.event not like '%rdbms%'

and s.username is not null

order by se.wait\_time;

**Find The Main Database Wait Events In A Particular Time Interval**

First determine the snapshot id values for the period in question.

In this example we need to find the SNAP\_ID for the period 10 PM to 11 PM on the 14th of November, 2012.

select snap\_id,begin\_interval\_time,end\_interval\_time

from dba\_hist\_snapshot

where to\_char(begin\_interval\_time,'DD-MON-YYYY')='14-NOV-2012'

and EXTRACT(HOUR FROM begin\_interval\_time) between 22 and 23;

set verify off  
select \* from (  
select active\_session\_history.event,  
sum(active\_session\_history.wait\_time +  
active\_session\_history.time\_waited) ttl\_wait\_time  
from dba\_hist\_active\_sess\_history active\_session\_history  
where event is not null  
and SNAP\_ID between &ssnapid and &esnapid  
group by active\_session\_history.event  
order by 2 desc)  
where rownum

**Top CPU Consuming SQL During A Certain Time Period**

Note – in this case we are finding the Top 5 CPU intensive SQL statements executed between 9.00 AM and 11.00 AM

select \* from (

select

SQL\_ID,

sum(CPU\_TIME\_DELTA),

sum(DISK\_READS\_DELTA),

count(\*)

from

DBA\_HIST\_SQLSTAT a, dba\_hist\_snapshot s

where

s.snap\_id = a.snap\_id

and s.begin\_interval\_time > sysdate -1

and EXTRACT(HOUR FROM S.END\_INTERVAL\_TIME) between 9 and 11

group by

SQL\_ID

order by

sum(CPU\_TIME\_DELTA) desc)

where rownum

**Which Database Objects Experienced the Most Number of Waits in the Past One Hour**

set linesize 120

col event format a40

col object\_name format a40

select \* from

(

select dba\_objects.object\_name,

dba\_objects.object\_type,

active\_session\_history.event,

sum(active\_session\_history.wait\_time +

active\_session\_history.time\_waited) ttl\_wait\_time

from v$active\_session\_history active\_session\_history,

dba\_objects

where

active\_session\_history.sample\_time between sysdate - 1/24 and sysdate

and active\_session\_history.current\_obj# = dba\_objects.object\_id

group by dba\_objects.object\_name, dba\_objects.object\_type, active\_session\_history.event

order by 4 desc)

where rownum < 6;

**Top Segments ordered by Physical Reads**

col segment\_name format a20

col owner format a10

select segment\_name,object\_type,total\_physical\_reads

from ( select owner||'.'||object\_name as segment\_name,object\_type,

value as total\_physical\_reads

from v$segment\_statistics

where statistic\_name in ('physical reads')

order by total\_physical\_reads desc)

where rownum

**Top 5 SQL statements in the past one hour**

select \* from (

select active\_session\_history.sql\_id,

dba\_users.username,

sqlarea.sql\_text,

sum(active\_session\_history.wait\_time +

active\_session\_history.time\_waited) ttl\_wait\_time

from v$active\_session\_history active\_session\_history,

v$sqlarea sqlarea,

dba\_users

where

active\_session\_history.sample\_time between sysdate - 1/24 and sysdate

and active\_session\_history.sql\_id = sqlarea.sql\_id

and active\_session\_history.user\_id = dba\_users.user\_id

group by active\_session\_history.sql\_id,sqlarea.sql\_text, dba\_users.username

order by 4 desc )

where rownum

**SQL with the highest I/O in the past one day**

select \* from

(

SELECT /\*+LEADING(x h) USE\_NL(h)\*/

h.sql\_id

, SUM(10) ash\_secs

FROM dba\_hist\_snapshot x

, dba\_hist\_active\_sess\_history h

WHERE x.begin\_interval\_time > sysdate -1

AND h.SNAP\_id = X.SNAP\_id

AND h.dbid = x.dbid

AND h.instance\_number = x.instance\_number

AND h.event in ('db file sequential read','db file scattered read')

GROUP BY h.sql\_id

ORDER BY ash\_secs desc )

where rownum

**Top CPU consuming queries since past one day**

select \* from (

select

SQL\_ID,

sum(CPU\_TIME\_DELTA),

sum(DISK\_READS\_DELTA),

count(\*)

from

DBA\_HIST\_SQLSTAT a, dba\_hist\_snapshot s

where

s.snap\_id = a.snap\_id

and s.begin\_interval\_time > sysdate -1

group by

SQL\_ID

order by

sum(CPU\_TIME\_DELTA) desc)

where rownum

**Find what the top SQL was at a particular reported time of day**

First determine the snapshot id values for the period in question.

In thos example we need to find the SNAP\_ID for the period 10 PM to 11 PM on the 14th of November, 2012.

select snap\_id,begin\_interval\_time,end\_interval\_time

from dba\_hist\_snapshot

where to\_char(begin\_interval\_time,'DD-MON-YYYY')='14-NOV-2012'

and EXTRACT(HOUR FROM begin\_interval\_time) between 22 and 23;

select \* from

(

select

sql.sql\_id c1,

sql.buffer\_gets\_delta c2,

sql.disk\_reads\_delta c3,

sql.iowait\_delta c4

from

dba\_hist\_sqlstat sql,

dba\_hist\_snapshot s

where

s.snap\_id = sql.snap\_id

and

s.snap\_id= &snapid

order by

c3 desc)

where rownum < 6

/

**Analyse a particular SQL ID and see the trends for the past day**

select

s.snap\_id,

to\_char(s.begin\_interval\_time,'HH24:MI') c1,

sql.executions\_delta c2,

sql.buffer\_gets\_delta c3,

sql.disk\_reads\_delta c4,

sql.iowait\_delta c5,

sql.cpu\_time\_delta c6,

sql.elapsed\_time\_delta c7

from

dba\_hist\_sqlstat sql,

dba\_hist\_snapshot s

where

s.snap\_id = sql.snap\_id

and s.begin\_interval\_time > sysdate -1

and

sql.sql\_id='&sqlid'

order by c7

/

**Do we have multiple plan hash values for the same SQL ID – in that case may be changed plan is causing bad performance**

select

SQL\_ID

, PLAN\_HASH\_VALUE

, sum(EXECUTIONS\_DELTA) EXECUTIONS

, sum(ROWS\_PROCESSED\_DELTA) CROWS

, trunc(sum(CPU\_TIME\_DELTA)/1000000/60) CPU\_MINS

, trunc(sum(ELAPSED\_TIME\_DELTA)/1000000/60) ELA\_MINS

from DBA\_HIST\_SQLSTAT

where SQL\_ID in (

'&sqlid')

group by SQL\_ID , PLAN\_HASH\_VALUE

order by SQL\_ID, CPU\_MINS;

**Top 5 Queries for past week based on ADDM recommendations**

/\*

Top 10 SQL\_ID's for the last 7 days as identified by ADDM

from DBA\_ADVISOR\_RECOMMENDATIONS and dba\_advisor\_log

\*/

col SQL\_ID form a16

col Benefit form 9999999999999

select \* from (

select b.ATTR1 as SQL\_ID, max(a.BENEFIT) as "Benefit"

from DBA\_ADVISOR\_RECOMMENDATIONS a, DBA\_ADVISOR\_OBJECTS b

where a.REC\_ID = b.OBJECT\_ID

and a.TASK\_ID = b.TASK\_ID

and a.TASK\_ID in (select distinct b.task\_id

from dba\_hist\_snapshot a, dba\_advisor\_tasks b, dba\_advisor\_log l

where a.begin\_interval\_time > sysdate - 7

and a.dbid = (select dbid from v$database)

and a.INSTANCE\_NUMBER = (select INSTANCE\_NUMBER from v$instance)

and to\_char(a.begin\_interval\_time, 'yyyymmddHH24') = to\_char(b.created, 'yyyymmddHH24')

and b.advisor\_name = 'ADDM'

and b.task\_id = l.task\_id

and l.status = 'COMPLETED')

and length(b.ATTR4) > 1 group by b.ATTR1

order by max(a.BENEFIT) desc) where rownum < 6;