7/7/2023

15 DB Monitoring SQL Scripts for DBAs

Asfaw Gedamu

1. List of active sessions in the database

```
set echo off
set linesize 95
set head on
set feedback on
col sid head "Sid" form 9999 trunc
col serial# form 99999 trunc head "Ser#"
col username form a8 trunc
col osuser form a7 trunc
col machine form a20 trunc head "Client|Machine"
col program form a15 trunc head "Client|Program"
col login form a11
col "last call" form 9999999 trunc head "Last Call|In Secs"
col status form a6 trunc
select sid,serial#,substr(username,1,10) username,substr(osuser,1,10) osuser,
substr(program||module,1,15) program,substr(machine,1,22) machine,
to_char(logon_time,'ddMon hh24:mi') login,
last_call_et "last call",status
from gv$session where status='ACTIVE'
order by 1
/
```

2. Current tablespace usage:

```
set feedback off
set pagesize 70;
set linesize 2000
set head on
COLUMN Tablespace format a25 heading 'Tablespace Name'
COLUMN autoextensible format a11 heading 'AutoExtend'
COLUMN files_in_tablespace format 999 heading 'Files'
COLUMN total_tablespace_space format 99999999 heading 'TotalSpace'
COLUMN total_used_space format 99999999 heading 'UsedSpace'
COLUMN total_tablespace_free_space format 99999999 heading 'FreeSpace'
COLUMN total_used_pct format 9999 heading '%Used'
COLUMN total_free_pct format 9999 heading '%Free'
COLUMN max_size_of_tablespace format 99999999 heading 'ExtendUpto'
COLUM total_auto_used_pct format 999.99 heading 'Max%Used'
COLUMN total_auto_free_pct format 999.99 heading 'Max%Free'
WITH tbs_auto AS
(SELECT DISTINCT tablespace_name, autoextensible
FROM dba_data_files
WHERE autoextensible = 'YES'),
files AS
(SELECT tablespace_name, COUNT (*) tbs_files,
SUM (BYTES/1024/1024) total_tbs_bytes
FROM dba_data_files
GROUP BY tablespace_name),
fragments AS
(SELECT tablespace_name, COUNT (*) tbs_fragments,
SUM (BYTES)/1024/1024 total_tbs_free_bytes,
MAX (BYTES)/1024/1024 max_free_chunk_bytes
FROM dba_free_space
```

```
GROUP BY tablespace_name),
AUTOEXTEND AS
(SELECT tablespace_name, SUM (size_to_grow) total_growth_tbs
FROM (SELECT tablespace_name, SUM (maxbytes)/1024/1024 size_to_grow
FROM dba_data_files
WHERE autoextensible = 'YES'
GROUP BY tablespace_name
UNION
SELECT tablespace_name, SUM (BYTES)/1024/1024 size_to_grow
FROM dba_data_files
WHERE autoextensible = 'NO'
GROUP BY tablespace_name)
GROUP BY tablespace_name)
SELECT c.instance_name,a.tablespace_name Tablespace,
CASE tbs_auto.autoextensible
WHEN 'YES'
THEN 'YES'
ELSE 'NO'
END AS autoextensible,
files.tbs_files files_in_tablespace,
files.total_tbs_bytes total_tablespace_space,
(files.total_tbs_bytes - fragments.total_tbs_free_bytes
) total_used_space,
fragments.total_tbs_free_bytes total_tablespace_free_space,
round(( ( (files.total_tbs_bytes - fragments.total_tbs_free_bytes)
/ files.total_tbs_bytes
)
* 100
)) total_used_pct,
round(((fragments.total_tbs_free_bytes / files.total_tbs_bytes) * 100
)) total_free_pct
FROM dba_tablespaces a,v$instance c , files, fragments, AUTOEXTEND, tbs_auto
WHERE a.tablespace_name = files.tablespace_name
AND a.tablespace_name = fragments.tablespace_name
AND a.tablespace_name = AUTOEXTEND.tablespace_name
AND a.tablespace_name = tbs_auto.tablespace_name(+)
order by total_free_pct;
```

3. Find the blocking session details:

```
SELECT
s.inst_id,
s.blocking_session,
s.sid,
s.serial#,
s.seconds_in_wait
FROM
gv$session s
WHERE
blocking_session IS NOT NULL;
```

4. Monitor TEMP tablespace usage:

```
select a.tablespace_name tablespace,
d.TEMP_TOTAL_MB,
sum (a.used_blocks * d.block_size) / 1024 / 1024 TEMP_USED_MB,
d.TEMP_TOTAL_MB - sum (a.used_blocks * d.block_size) / 1024 / 1024
TEMP_FREE_MB
from v$sort_segment a,
(
select b.name, c.block_size, sum (c.bytes) / 1024 / 1024 TEMP_TOTAL_MB
from v$tablespace b, v$tempfile c
where b.ts#= c.ts#
group by b.name, c.block_size
) d
where a.tablespace_name = d.name
group by a.tablespace_name, d.TEMP_TOTAL_MB;
```

5. Find the long running queries:

```
select sid,inst_id,opname,totalwork,sofar,start_time,time_remaining
from gv$session_longops
where totalwork<>sofar
/
```

6. Get os spid from sid:

```
set lines 123
col USERNAME for a15
col OSUSER for a8
col MACHINE for a15
col PROGRAM for a20
select b.spid, a.username, a.program , a.osuser ,a.machine, a.sid, a.serial#,
a.status from gv$session a, gv$process b
where addr=paddr(+) and sid=&sid;
```

7. Get sid from os spid:

```
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='&spid'
order by b.spid;
```

8. Monitor undo tablespace usage:

```
select a.tablespace_name, SIZEMB, USAGEMB, (SIZEMB - USAGEMB) FREEMB
```

```
from (select sum(bytes) / 1024 / 1024 SIZEMB, b.tablespace_name
from dba_data_files a, dba_tablespaces b
where a.tablespace_name = b.tablespace_name
and b.contents = 'UNDO'
group by b.tablespace_name) a,
(select c.tablespace_name, sum(bytes) / 1024 / 1024 USAGEMB
from DBA_UNDO_EXTENTS c
where status <> 'EXPIRED'
group by c.tablespace_name) b
where a.tablespace_name = b.tablespace_name;
```

9. Get sql_text of an SID:

```
col sql_text form a80
set lines 120
select sql_text from gv$sqltext where hash_value=
(select sql_hash_value from gv$session where sid=&1 and inst_id=&inst_id)
order by piece
/
```

10. Locks present in the database

```
col session_id head 'Sid' form 9999
col object_name head "Table|Locked" form a30
col oracle_username head "Oracle|Username" form a10 truncate
col os_user_name head "OS|Username" form a10 truncate
col process head "Client|Process|ID" form 99999999
col mode_held form a15
select lo.session_id,lo.oracle_username,lo.os_user_name,
lo.process,do.object_name,
decode(lo.locked_mode,0, 'None',1, 'Null',2, 'Row Share (SS)',
3, 'Row Excl (SX)',4, 'Share',5, 'Share Row Excl (SSX)',6, 'Exclusive',
to_char(lo.locked_mode)) mode_held
from v$locked_object lo, dba_objects do
where lo.object_id = do.object_id
order by 1,5
/
```

11. Find the sessions generating lot of redo:

```
set lines 2000
set pages 1000
col sid for 99999
col name for a09
col username for a14
col PROGRAM for a21
col MODULE for a25
select s.sid,sn.SERIAL#,n.name, round(value/1024/1024,2) redo_mb,
sn.username,sn.status,substr (sn.program,1,21) "program", sn.type,
sn.module,sn.sql_id
from v$sesstat s join v$statname n on n.statistic# = s.statistic#
```

```
join v$session sn on sn.sid = s.sid where n.name like 'redo size' and
s.value!=0 order by
redo_mb desc;
```

12. Find the session generating undo data:

```
select a.sid, a.serial#, a.username, b.used_urec used_undo_record,
b.used_ublk used_undo_blocks
from v$session a, v$transaction b
where a.saddr=b.ses_addr ;
```

13. Find temp usage of the sessions:

```
SELECT b.tablespace,
ROUND(((b.blocks*p.value)/1024/1024),2)||'M' AS temp_size,
a.inst_id as Instance,
a.sid||','||a.serial# AS sid_serial,
NVL(a.username, '(oracle)') AS username,
a.program,
a.status,
a.sql_id
FROM gv$session a,
gv$sort_usage b,
gv$parameter p
WHERE p.name = 'db_block_size'
AND a.saddr = b.session_addr
AND a.inst_id=b.inst_id
AND a.inst_id=p.inst_id
ORDER BY temp_size desc
/
```

14. Monitor rollback transactions:

```
select state,UNDOBLOCKSDONE,UNDOBLOCKSTOTAL,
UNDOBLOCKSDONE/UNDOBLOCKSTOTAL*100
from gv$fast_start_transactions;
```

15. Top queries with high elapsed_time:

— Queries in last 1 hour (Run from Toad, for proper view)

```
Select
module,parsing_schema_name,inst_id,sql_id,CHILD_NUMBER,sql_plan_baseline,sql_
profile,plan_hash_value,sql_fulltext,
to_char(last_active_time,'DD/MM/YY HH24:MI:SS' ),executions,
elapsed_time/executions/1000/1000,
rows_processed,sql_plan_baseline from gv$sql where last_active_time>sysdate-
1/24
and executions <> 0 order by elapsed_time/executions desc
```