ORACLE-LEARN

Welcome to our oracle learning world !

Monitoring shared pool usage

Posted on July 4, 2014

Monitoring shared pool usage

–SHARED POOL QUICK CHECK NOTES:

```
select 'You may need to increase the SHARED_POOL_RESERVED_SIZE' Descri
'Request Failures = '||REQUEST_FAILURES Logic
from v$shared_pool_reserved
where REQUEST_FAILURES > 0
and 0 != (select to_number(VALUE) from v$parameter where NAME = 'share
union
select 'You may be able to decrease the SHARED_POOL_RESERVED_SIZE' Des
'Request Failures = '||REQUEST_FAILURES Logic
from v$shared_pool_reserved where REQUEST_FAILURES < 5
and 0 != ( select to_number(VALUE) from v$parameter where NAME = 'shar
```

–SHARED POOL MEMORY USAGE NOTES:>

V$DB_OBJECT_CACHE

This view displays database objects that are cached in the library cache. Objects include tables, indexes,

clusters, synonym definitions, PL/SQL procedures and packages, and triggers.

```
select OWNER, NAME||' - '||TYPE object, SHARABLE_MEM
from v$db_object_cache
where SHARABLE_MEM > 10000
and type in ('PACKAGE','PACKAGE BODY','FUNCTION','PROCEDURE')
order by SHARABLE_MEM desc
```

–LOADS INTO SHARED POOL NOTES:

select OWNER, NAME||' – '||TYPE object, LOADS

```
from v$db_object_cache
where LOADS > 3
and type in (‘PACKAGE’,’PACKAGE BODY’,’FUNCTION’,’PROCEDURE’)
order by LOADS desc
```

–SHARED POOL EXECUTION NOTES:

```
select OWNER, NAME||’ – ‘||TYPE object, EXECUTIONS
from v$db_object_cache
where EXECUTIONS > 100
and type in (‘PACKAGE’,’PACKAGE BODY’,’FUNCTION’,’PROCEDURE’)
order by EXECUTIONS desc
```

–SHARED POOL DETAIL NOTES:

```
select OWNER, NAME, DB_LINK, NAMESPACE, TYPE, SHARABLE_MEM,
LOADS, EXECUTIONS, LOCKS, PINS
from v$db_object_cache
order by OWNER, NAME
```

–SHARED POOL V$LIBRARYCACHE STATISTIC NOTES:

```
select NAMESPACE, GETS, GETHITS, round(GETHITRATIO*100,2) gethit_ratio,
PINS, PINHITS, round(PINHITRATIO*100,2) pinhit_ratio, RELOADS, INVALIDATIONS
from v$librarycache
```

–SHARED POOL RESERVED SIZE NOTES:

```
select NAME, VALUE
from v$parameter
where NAME like ‘%reser%’
```

–PINNED OBJECT NOTES:

```
select NAME,TYPE,KEPT
from v$db_object_cache
where KEPT = ‘YES’
```

Script to Estimate Shared Pool Utilization

```
This script estimates Shared Pool utilization.

Sample Output

Copy Script to Clipboard

REM LOCATION:   Database TuningShared Pool Reports
REM FUNCTION:   Estimates shared pool utilization
REM TESTED ON:  7.3.3.5, 8.0.4.1, 8.1.5, 8.1.7, 9.0.1, 10.2.0.3, 11.1.
REM PLATFORM:   non-specific
REM REQUIRES:   v$db_object_cache, v$sqlarea, v$sesstat, v$statname,
REM             v$sgastat, v$parameter
REM
REM  This is a part of the Knowledge Xpert for Oracle Administration l
REM  Copyright (C) 2008 Quest Software
REM  All rights reserved.
REM
REM******************** Knowledge Xpert for Oracle Administration ****
REM
REM NOTES:     Based on current database usage. This should be
REM            run during peak operation, after all stored
REM            objects i.e. packages, views have been loaded.
REM
REM 08/02/08 Robert Freeman - Modified to use v$sgastat instead v$para
REM                           shared pool size.
REM*******************************************************************
REM

REM If running Shared Server uncomment the mts calculation and output
SET serveroutput on;

DECLARE
   object_mem       NUMBER;
   shared_sql       NUMBER;
   cursor_mem       NUMBER;
   mts_mem          NUMBER;
   used_pool_size   NUMBER;
   free_mem         NUMBER;
   pool_size        VARCHAR2 (512);                     -- Now from V$
BEGIN
   -- Stored objects (packages, views)
   SELECT SUM (sharable_mem)
     INTO object_mem
     FROM v$db_object_cache;

   -- Shared SQL -- need to have additional memory if dynamic SQL used
   SELECT SUM (sharable_mem)
     INTO shared_sql
     FROM v$sqlarea;

   -- User Cursor Usage -- run this during peak usage.
   --  assumes 250 bytes per open cursor, for each concurrent user.
   SELECT SUM (250 * users_opening)
     INTO cursor_mem
     FROM v$sqlarea;

   -- For a test system -- get usage for one user, multiply by # users
   -- select (250 * value) bytes_per_user
   -- from v$sesstat s, v$statname n
   -- where s.statistic# = n.statistic#
   -- and n.name = 'opened cursors current'
   -- and s.sid = 25;  -- where 25 is the sid of the process
   -- MTS memory needed to hold session information for shared server
   -- This query computes a total for all currently logged on users (r
   --  multiply by # users.
   SELECT SUM (VALUE)
     INTO mts_mem
     FROM v$sesstat s, v$statname n
    WHERE s.statistic# = n.statistic# AND n.NAME = 'session uga memory

   -- Free (unused) memory in the SGA: gives an indication of how much
   -- is being wasted out of the total allocated.
   SELECT BYTES
     INTO free_mem
     FROM v$sgastat
    WHERE NAME = 'free memory' AND pool = 'shared pool';

   -- For non-MTS add up object, shared sql, cursors and 20% overhead.
   used_pool_size := ROUND (1.2 * (object_mem + shared_sql + cursor_me

   -- For MTS mts contribution needs to be included (comment out previ
   -- used_pool_size := round(1.2*(object_mem+shared_sql+cursor_mem+mt
   SELECT SUM (BYTES)
     INTO pool_size
     FROM v$sgastat
    WHERE pool = 'shared pool';

   -- Display results
   DBMS_OUTPUT.put_line ('Shared Pool Memory Utilization Report');
   DBMS_OUTPUT.put_line ('Obj mem:  ' || TO_CHAR (object_mem) || ' byt
   DBMS_OUTPUT.put_line ('Shared sql:  ' || TO_CHAR (shared_sql) || '
   DBMS_OUTPUT.put_line ('Cursors:  ' || TO_CHAR (cursor_mem) || ' byt
   -- dbms_output.put_line ('MTS session: '||to_char (mts_mem) || ' by
   DBMS_OUTPUT.put_line (   'Free memory: '
                         || TO_CHAR (free_mem)
                         || ' bytes '
                         || '('
                         || TO_CHAR (ROUND (free_mem / 1024 / 1024, 2)
                         || 'MB)'
                        );
   DBMS_OUTPUT.put_line (   'Shared pool utilization (total):  '
                         || TO_CHAR (used_pool_size)
                         || ' bytes '
                         || '('
                         || TO_CHAR (ROUND (used_pool_size / 1024 / 10
                         || 'MB)'
                        );
```

```
      DBMS_OUTPUT.put_line (   'Shared pool allocation (actual):  '
                             || pool_size
                             || ' bytes '
                             || '('
                             || TO_CHAR (ROUND (pool_size / 1024 / 1024, 2
                             || 'MB)'
                            );
      DBMS_OUTPUT.put_line (   'Percentage Utilized:  '
                             || TO_CHAR (ROUND (used_pool_size / pool_size
                            );
   END;
   /
```

Sample Output

Shared Pool Memory Utilization Report

Obj mem: 64949920 bytes

Shared sql: 21737236 bytes

Cursors: 14250 bytes

Free memory: 54037748 bytes (51.53MB)

Shared pool utilization (total): 104041687 bytes (99.22MB)

Shared pool allocation (actual): 180359768 bytes (172MB)

Percentage Utilized: 58

Shared Pool Reports

- Script to Report Open Cursors Per User
- Script to Report Data Dictionary Cache Condition
- Script to Estimate Shared Pool Utilization
- Script to Report on the Library Cache
- Script to Report on the v$rowcache Table
- Script to Estimate Shared Pool Sizing Change Impacts

Twitter Facebook LinkedIn Telegram WhatsApp Skype

More

Like

Be the first to like this.

RELATED

Monitoring Buffer Pool size usage
July 4, 2014
In "Oracle Performance

Transporting FULL Database from 11g Database to 12c Database
January 18, 2016

Flushing a Single SQL Statement out of the Object Library Cache
February 6, 2014

Tuning"

In "Oracle Upgrade"

In "Oracle Performance Tuning"

This entry was posted in **Oracle Performance Tuning** and tagged **Monitoring shared pool usage** by **hiteshgondalia**. Bookmark the **permalink [https://hiteshgondalia.wordpress.com/2014/07/04/monitoring-shared-pool-usage/]** .