Troubleshooting Database Contention With V$Wait_Chains (Doc ID 1428210.1)

To Bottom

In this Document

APPLIES TO:

Oracle Database Backup Service - Version N/A and later
Oracle Database Exadata Express Cloud Service - Version N/A and later
Oracle Database Cloud Service - Version N/A and later
Oracle Database - Enterprise Edition - Version 11.1.0.6 to 12.1.0.2 [Release 11.1 to 12.1]
Oracle Database Cloud Schema Service - Version N/A and later
Information in this document applies to any platform.

PURPOSE

The purpose of this document is to provide a fast and easy way to troubleshoot database contention and hangs via SQL in version 11g with v$wait_chains.

TROUBLESHOOTING STEPS

Introduction

Starting in 11g release 1, the dia0 background processes starts collecting hanganalyze information and stores this in memory in the "hang analysis cache". It does this every 3 seconds for local hanganalyze information and every 10 seconds for global (RAC) hanganalyze information.

The data stored in the "hang analysis cache" can be extremely valuable for troubleshooting live database contention and hangs.

There are many database features that utilize the hang analysis cache for example: Hang Management, Resource Manager Idle Blocker Kill, SQL Tune Hang Avoidance and PMON cleanup as well as external tools that use the hang analysis cache like Procwatcher.

The following is a describe of v$wait_chains in 11g R2:

```
SQL> desc v$wait_chains
  Name Null? Type
  ----------------------------------------- -------- ----------------------
  CHAIN_ID NUMBER
  CHAIN_IS_CYCLE VARCHAR2(5)
  CHAIN_SIGNATURE VARCHAR2(801)
  CHAIN_SIGNATURE_HASH NUMBER
  INSTANCE NUMBER
  OSID VARCHAR2(25)
  PID NUMBER
  SID NUMBER
  SESS_SERIAL# NUMBER
  BLOCKER_IS_VALID VARCHAR2(5)
  BLOCKER_INSTANCE NUMBER
  BLOCKER_OSID VARCHAR2(25)
  BLOCKER_PID NUMBER
  BLOCKER_SID NUMBER
  BLOCKER_SESS_SERIAL# NUMBER
  BLOCKER_CHAIN_ID NUMBER
  IN_WAIT VARCHAR2(5)
  TIME_SINCE_LAST_WAIT_SECS NUMBER
  WAIT_ID NUMBER
  WAIT_EVENT NUMBER
  WAIT_EVENT_TEXT VARCHAR2(64)
  P1 NUMBER
  P1_TEXT VARCHAR2(64)
  P2 NUMBER
```

```
  P2_TEXT VARCHAR2(64)
  P3 NUMBER
  P3_TEXT VARCHAR2(64)
  IN_WAIT_SECS NUMBER
  TIME_REMAINING_SECS NUMBER
  NUM_WAITERS NUMBER
  ROW_WAIT_OBJ# NUMBER
  ROW_WAIT_FILE# NUMBER
  ROW_WAIT_BLOCK# NUMBER
  ROW_WAIT_ROW# NUMBER
```

request"<="gc buffer busy acquire"<="library cache lock" Waits on a Single Session [2546022.1]

Database hang due to update user$ set spare6=DECODE(to_char(:2, "YYYY-MM-DD"), "0000-00-00", to_date(NULL), :2) where user#=:1 [2782511.1]

Show More

Note: There is no gv$ equivalent as v$wait_chains would report on multiple instances in a multi-instance (RAC) environment.

Useful SQLs for querying v$wait_chains:

Basic Information

```
SQL> SELECT chain_id, num_waiters, in_wait_secs, osid, blocker_osid, substr(wait_event_text,1,30)
 FROM v$wait_chains; 2

 CHAIN_ID NUM_WAITERS IN_WAIT_SECS OSID BLOCKER_OSID SUBSTR(WAIT_EVENT_TEXT,1,30)
 ---------- ----------- ------------ ------------------------ ------------------------ --------------
----------------
 1 0 10198 21045 21044 enq: TX - row lock contention
 1 1 10214 21044 SQL*Net message from client
```

Additional Information (formatted) - Top 100 wait chain processes

```
 set pages 1000
 set lines 120
 set heading off
 column w_proc format a50 tru
 column instance format a20 tru
 column inst format a28 tru
 column wait_event format a50 tru
 column p1 format a16 tru
 column p2 format a16 tru
 column p3 format a15 tru
 column Seconds format a50 tru
 column sincelw format a50 tru
 column blocker_proc format a50 tru
 column waiters format a50 tru
 column chain_signature format a100 wra
 column blocker_chain format a100 wra

 SELECT *
 FROM (SELECT 'Current Process: '||osid W_PROC, 'SID '||i.instance_name INSTANCE,
 'INST #: '||instance INST,'Blocking Process: '||decode(blocker_osid,null,'<none>',blocker_osid)||
 ' from Instance '||blocker_instance BLOCKER_PROC,'Number of waiters: '||num_waiters waiters,
 'Wait Event: ' ||wait_event_text wait_event, 'P1: '||p1 p1, 'P2: '||p2 p2, 'P3: '||p3 p3,
 'Seconds in Wait: '||in_wait_secs Seconds, 'Seconds Since Last Wait: '||time_since_last_wait_secs
sincelw,
 'Wait Chain: '||chain_id ||': '||chain_signature chain_signature,'Blocking Wait Chain:
'||decode(blocker_chain_id,null,
 '<none>',blocker_chain_id) blocker_chain
 FROM v$wait_chains wc,
 v$instance i
 WHERE wc.instance = i.instance_number (+)
 AND ( num_waiters > 0
 OR ( blocker_osid IS NOT NULL
 AND in_wait_secs > 10 ) )
 ORDER BY chain_id,
 num_waiters DESC)
 WHERE ROWNUM < 101;
```

```
Current Process: 21549                          SID RAC1              INST #: 1
Blocking Process: <none> from Instance          Number of waiters: 1
Wait Event: SQL*Net message from client         P1: 1650815232    P2: 1             P3: 0
Seconds in Wait: 36                             Seconds Since Last Wait:
Wait Chain: 1: 'SQL*Net message from client'<='enq: TX - row lock contention'
Blocking Wait Chain: <none>

Current Process: 25627                          SID RAC1              INST #: 1
Blocking Process: 21549 from Instance 1         Number of waiters: 0
Wait Event: enq: TX - row lock contention       P1: 1415053318    P2: 524316        P3: 50874
Seconds in Wait: 22                             Seconds Since Last Wait:
Wait Chain: 1: 'SQL*Net message from client'<='enq: TX - row lock contention'
Blocking Wait Chain: <none>
```

ospid 25627 is waiting for a TX lock and is blocked by ospid 21549
ospid 21549 is idle waiting for "SQL*Net message from client"

Note: You can use Procwatcher proactively to monitor v$wait_chains and dumps diagnostic information if database contention is detected.

For information on Procwatcher see:
Document 459694.1 Procwatcher: Script to Monitor and Examine Oracle DB and Clusterware Processes

For an example of how to use Procwatcher to trap database contention problems see:
Document 1352623.1 How To Troubleshoot Database Contention With Procwatcher

Final Blocking Session in 11.2

In 11.2 you can also add v$session.final_blocking_session to see the final blocker. The final blocker is the session/process at the top of the wait chain. This is the session/process that maybe causing the problem. Example of query with final_blocking_session info:

```
set pages 1000
set lines 120
set heading off
column w_proc format a50 tru
column instance format a20 tru
column inst format a28 tru
column wait_event format a50 tru
column p1 format a16 tru
column p2 format a16 tru
column p3 format a15 tru
column Seconds format a50 tru
column sincelw format a50 tru
column blocker_proc format a50 tru
column fblocker_proc format a50 tru
column waiters format a50 tru
column chain_signature format a100 wra
column blocker_chain format a100 wra

SELECT *
FROM (SELECT 'Current Process: '||osid W_PROC, 'SID '||i.instance_name INSTANCE,
 'INST #: '||instance INST,'Blocking Process: '||decode(blocker_osid,null,'<none>',blocker_osid)||
 ' from Instance '||blocker_instance BLOCKER_PROC,
 'Number of waiters: '||num_waiters waiters,
 'Final Blocking Process: '||decode(p.spid,null,'<none>',
 p.spid)||' from Instance '||s.final_blocking_instance FBLOCKER_PROC,
 'Program: '||p.program image,
 'Wait Event: ' ||wait_event_text wait_event, 'P1: '||wc.p1 p1, 'P2: '||wc.p2 p2, 'P3: '||wc.p3 p3,
 'Seconds in Wait: '||in_wait_secs Seconds, 'Seconds Since Last Wait: '||time_since_last_wait_secs sincelw,
 'Wait Chain: '||chain_id ||': '||chain_signature chain_signature,'Blocking Wait Chain: '||decode(blocker_chain_id,null,
 '<none>',blocker_chain_id) blocker_chain
FROM v$wait_chains wc,
 gv$session s,
 gv$session bs,
 gv$instance i,
 gv$process p
WHERE wc.instance = i.instance_number (+)
 AND (wc.instance = s.inst_id (+) and wc.sid = s.sid (+)
 and wc.sess_serial# = s.serial# (+))
 AND (s.final_blocking_instance = bs.inst_id (+) and s.final_blocking_session = bs.sid (+))
 AND (bs.inst_id = p.inst_id (+) and bs.paddr = p.addr (+))
 AND ( num_waiters > 0
 OR ( blocker_osid IS NOT NULL
 AND in_wait_secs > 10 ) )
ORDER BY chain_id,
 num_waiters DESC)
WHERE ROWNUM < 101;
```

```
Current Process: 2309                          SID RAC1              INST #: 1
Blocking Process: <none> from Instance         Number of waiters: 2
Final Blocking Process: <none> from Instance   Program:
Wait Event: SQL*Net message from client        P1: 1650815232    P2: 1            P3: 0
Seconds in Wait: 157                           Seconds Since Last Wait:
Wait Chain: 1: 'SQL*Net message from client'<='enq: TM - contention'<='enq: TM - contention'
Blocking Wait Chain: <none>

Current Process: 2395                          SID RAC1              INST #: 1
Blocking Process: 2309 from Instance 1         Number of waiters: 1
Final Blocking Process: 2309 from Instance 1   Program: oracle@[redacted].oracle.com (TNS V1-V3)
Wait Event: enq: TM - contention               P1: 1414332420    P2: 73480        P3: 0
Seconds in Wait: 139                           Seconds Since Last Wait:
Wait Chain: 1: 'SQL*Net message from client'<='enq: TM - contention'<='enq: TM - contention'
Blocking Wait Chain: <none>
```

Here we can see that ospid 2309 is the final blocker.

Example of Output of Query 1 using Oracle Enterprise Manager

Last Executed SQL

```
select chain_id, num_waiters, in_wait_secs, osid,
blocker_osid, wait_event_text
from v$wait_chains
```

Last Execution Details

SQL Repair Advisor | SQL Details | Schedule SQL Tuning Advisor

Results | Statistics | Plan

Execution Time (seconds) **0.082**

CHAIN_ID	NUM_WAITERS	IN_WAIT_SECS	OSID	BLOCKER_OSID	WAIT_EVENT_TEXT
1	0	85	1135	454	enq: TM - contention
1	3	156	454	507	enq: TM - contention
1	4	3	507		SQL*Net message from client
2	0	141	615	454	enq: TM - contention
3	0	91	1111	454	enq: TM - contention
4	0	3	29375		Streams AQ: waiting for messages in the queue

SQL Repair Advisor | SQL Details | Schedule SQL Tuning Advisor

In the above example we see that:

- ospid: 1135 is waiting for a TM enqueue and is waiting for ospid: 454.
- ospid: 454 is waiting for a TM enqueue and is waiting for ospid: 507.
- ospid: 507 is idle waiting for "SQL*Net message from client".

Also note that ospid's 615 and 1111 are also waiting for the first wait chain meaning that ospid: 507 is ultimately blocking these 2 processes as well.

Solutions to this contention would include trying to get ospid: 507 release the lock or even killing the session or process if there is no other alternative.

Attachments

- wait_chains.PNG (22.42 KB)
- wait_chains2.PNG (198.65 KB)
- Wait Chains Query 2 Output (18.41 KB)
- Wait Chains 4 (19.89 KB)

Related

Products

- Oracle Cloud > Oracle Platform Cloud > Oracle Database Backup Service > Oracle Database Backup Service
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Service > Oracle Database Exadata Express Cloud Service
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Service > Oracle Database Cloud Service
- Oracle Database Products > Oracle Database Suite > Oracle Database > Oracle Database - Enterprise Edition > RDBMS > Database Level Performance Issues (not SQL Tuning)
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Service > Oracle Database Cloud Schema Service
- Oracle Cloud > Oracle Infrastructure Cloud > Oracle Cloud at Customer > Gen 1 Exadata Cloud at Customer (Oracle Exadata Database Cloud Machine)
- Oracle Cloud > Oracle Platform Cloud > Oracle Cloud Infrastructure - Database Service > Oracle Cloud Infrastructure - Database Service
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Exadata Service > Oracle Database Cloud Exadata Service

Keywords

TROUBLESHOOT

Translations

- English Source
- Japanese 日本語

Back to Top