|  |  |  |  |
| --- | --- | --- | --- |
|  | **Troubleshooting "Global Enqueue Services Deadlock detected" (Doc ID 1443482.1)** | [To Bottom](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254)  [To Bottom](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254) |  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In this Document**   |  |  | | --- | --- | |  | [Purpose](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#PURPOSE) | |  | [Troubleshooting Steps](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#TRBLSHOOT) | |  |  |  |  | | --- | --- | --- | |  | [1. TX deadlock in Exclusive(X) mode](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section21) | | |  | [2. TX deadlock in Share(S) mode](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section22) |  |  |  | | --- | --- | |  | [3. TM deadlock](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section23) | |  | [4. Single resource deadlock for TX , TM or IV](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section24) | |  |  |  | | --- | --- | |  | [5. LB deadlock](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section25) | |  | [6. Known Issues](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section26) | |  |  |  |  | | --- | --- | --- | |  | [7. Further Diagnosis](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#aref_section27) | | |  | [References](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=nfgg1wd1h_4&_afrLoop=263334775117254#REF) |     **Applies to:**  Oracle Database - Enterprise Edition - Version 10.1.0.2 and later  Information in this document applies to any platform.  **Purpose**  This note is to provide some common causes and solutions for message "Global Enqueue Services Deadlock detected" reported in alert log.  **Troubleshooting Steps**  In single instance environment, when a deadlock happens, it often reports ORA-60, see [Document 15476.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=15476.1) FAQ: Detecting and Resolving Locking Conflicts and Ora-00060 errors. In a Real Application Cluster (RAC) environment, instead of ORA-60, one would see the following messages in database alert log:  Global Enqueue Services Deadlock detected. More info in file  /u01/diag/rdbms/rac/RAC1/trace/RAC1\_ora\_3457040.trc.  OR   Global Enqueue Services Deadlock detected. More info in file  /u01/diag/rdbms/rac/RAC1/trace/RAC1\_lmd0\_30429.trc.  The major difference of deadlock between single instance and RAC is the sessions involved in a deadlock could be from different instances and there could be more than 2 sessions involved. When Global Enqueue Service Deadlock is reported, the session which initiates the deadlock checking will be terminated to resolve the deadlock. There are different deadlock types in RAC environment, many are similar to single instance deadlock.  To understand the basics of deadlock, refer to the following documents:  [Document 62365.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=62365.1) What to do with "ORA-60 Deadlock Detected" Errors [Document 62354.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=62354.1) TX Transaction locks - Example wait scenarios  GV$GES\_ENQUEUE and GV$GES\_BLOCKING\_ENQUEUE can be used to query Global Enqueue Service(GES) lock, they may not necessary involve in a deadlock. Deadlock related information (session, SQL statement etc) will be printed in lmd0 or foreground trace file.  With [Bug 6343023](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=1443482.1&id=6343023) fixed in 10.2.0.5, 11.1.0.7 and all 11.2, the offending SQL statements from involved sessions will be written in the trace. For earlier version or if there is no SQL statements in the trace, apply [patch 6343023](https://support.oracle.com/epmos/faces/ui/patch/PatchDetail.jspx?parent=DOCUMENT&sourceId=1443482.1&patchId=6343023) or use step 6 to gather system state dump.       Here are some common deadlock types:  ***1. TX deadlock in Exclusive(X) mode***  trace shows:   Global Wait-For-Graph(WFG) at ddTS[0.170] :  BLOCKED 0x8aafb0ec **5** wq 2 cvtops x1 **TX** 0x320001.0x121c97 [99000-0001-00000002]**0  BLOCKER 0x8aafafec 5** wq 1 cvtops x8 **TX** 0x320001.0x121c97 [9A000-0001-00000002]**0  BLOCKED 0x8acb55e4 5** wq 2 cvtops x1 **TX** 0x430003.0x3843f [9A000-0001-00000002] **0  BLOCKER 0x8acb54e4 5** wq 1 cvtops x8 **TX** 0x430003.0x3843f [99000-0001-00000002] **0   These values are:  <BLOCKED|BLOCKER> <lockp> <cvt|held mode>** <which q> <convert options> **<res name - type id1.id2>** [deadlock-level-trace] **<node>**    Above graph shows the deadlock is related with 2 TX lock in mode 5 (exclusive) which happens on instance 1 (instance# starts from 0)  Above deadlock means two sessions involved in TX-0x320001-0x121c97 and TX-0x430003-0x3843f forms a deadlock, both sessions are from instance 1.   This is a typical application transaction TX enqueue lock, usually caused by SQL commit sequence and high concurrency. To avoid such deadlock, application code and logic need to be modified.  The application and SQL involved in the deadlock can be found in lmd0 or foreground trace (check all instances). Search for: "user session for deadlock lock" section to find out the SQL involved in the deadlock. For example:  user session for deadlock lock 0x8aafb0ec:  ...    current SQL:    update test set OWNER='APPS' where rownum < 2    ***2. TX deadlock in Share(S) mode***  trace shows:   Global Wait-For-Graph(WFG) at ddTS[0.b7] :  BLOCKED 0x2310e8918 **3** [0x731000e][0x56268],[**TX**] [F1000-0001-0000000F] 0  BLOCKER 0x2310e87c8 **3** [0x731000e][0x56268],[**TX**] [19F000-0001-00000011] 0  BLOCKED 0x2310e3c50 **3** [0x72a0023][0x530d7],[**TX**] [19F000-0001-00000011] 0  BLOCKER 0x2310e7e80 **3** [0x72a0023][0x530d7],[**TX**] [F1000-0001-0000000F] 0    Corresponding to:  <BLOCKED|BLOCKER> <lockp> <cvt|held mode> <res name> <pid|did|txn\_id> <node>   mode 3 is shared lock  The causes for TX deadlock in S mode wait can be:  a. ITL contention, eg: INITRANS setting for the object is too small, it can not handle the number of concurrent transactions.  The solution is to increase INITRANS setting for the object involved in the deadlock using "alter table" or "alter index" command  The SQL involved in the deadlock can be found in lmd0 or client trace. The object involved in the SQL should be checked including table and its associated index.  b. If the object involved is an unique key index, the wait could be caused by uniqueness validation. Application needs to be checked to avoid unique key violation.  c. If the object involved has a bitmap index, then the bitmap index should be dropped to accommodate concurrent DML. Please refer to [Document 1496403.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=1496403.1) ORA-60 DEADLOCK DUE TO BITMAP INDEX IN RAC.  ***3. TM deadlock***  trace shows:   Global Wait-For-Graph(WFG) at ddTS[0.1] :  BLOCKED 0x7000003ccbf4798 3 wq 2 cvtops x1 **TM** 0x1cbde.0x0 [1004-004D-00000003] 0  BLOCKER 0x7000003d0bf9ad8 3 wq 1 cvtops x1 TM 0x1cbde.0x0 [200A-00AC-00000019] 1  BLOCKED 0x7000003d0bfcf88 2 wq 2 cvtops x1 TM 0x1cc77.0x0 [200A-00AC-00000019] 1  BLOCKER 0x7000003cc338e88 2 wq 2 cvtops x1 TM 0x1cc77.0x0 [2006-0063-00000055] 1  BLOCKED 0x7000003cc338e88 3 wq 2 cvtops x1 TM 0x1cc77.0x0 [2006-0063-00000055] 1  BLOCKER 0x7000003c879f9c0 3 wq 1 cvtops x1 TM 0x1cc77.0x0 [2006-0063-00000020] 1  BLOCKED 0x7000003c87978a8 2 wq 2 cvtops x1 TM 0x1cbde.0x0 [2006-0063-00000020] 1  BLOCKER 0x7000003ccbf4798 2 wq 2 cvtops x1 TM 0x1cbde.0x0 [1004-004D-00000003] 0  The object involved here are 0x1cbde and 0x1cc77, convert the hex number to decimal, they are the object\_id for the tables involved in above deadlock  The deadlock is usually caused by missing index for foreign key constraint, refer to [Document 473124.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=473124.1) - "Frequent GES: Potential Blocker (Pid=nnnn) On Resource TM-<id1>-<id2>" for more information. Check dba\_constraints and dba\_index to verify if foreign key index is missing. Also refer to [Document 1019527.6](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=1019527.6) Script to Check for Foreign Key Locking Issues for a Specific User which will generate a report for all problem objects.  The solution is to create index for every foreign key constraint.  ***4. Single resource deadlock for TX , TM or IV***  trace shows:   Single resource deadlock: blocking enqueue which blocks itself, f 1  Granted global enqueue 0xd078cec0  ...  resname :[0x2001f][0x1a96c3],[TX]  or  resname :[0x00001432][0x0],[TM]  or  resname : [0xbb7cc5db][0x82d0d4b5],[IV]  a. For single resource deadlock on TX enqueue, often it is caused by using autonomous transaction in stored procedure or PL/SQL. It is a known issue that the use of autonomous transactions is vulnerable to deadlocks. Please check out Oracle® Database Concepts  [Overview of Autonomous Transactions](http://docs.oracle.com/cd/E11882_01/server.112/e40540/transact.htm#CNCPT417) for detail explanation. Since AUTONOMOUS transaction has been used in the stored procedure, the system would consider any DML statement under this transaction as a separate one (commit/rollback won't affect the parent), and this would cause conflict if the same row is involved in the parent transaction (INSERT, UPDATE or DELETE), and hence deadlock is reported rightly. Usually the SQL involved in the deadlock is called from a stored procedure or PL/SQL with the following line:  **PRAGMA AUTONOMOUS\_TRANSACTION;**  To avoid such deadlock, please remove the autonomous transaction in the application code.  b. If there is no autonomous\_transaction involved, please check out [Document 6145177.8](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=6145177.8), it can also be caused by [Bug 6145177](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=1443482.1&id=6145177) - Single resource deadlock with a zero DID  c. For single resource deadlock on TM enqueue, missing foreign key index is often the cause, please check case 3 for the solution.  d. For single resource deadlock type IV (Instance Validation), refer to [Document 973178.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=973178.1), as mentioned in [Bug 8843816](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=1443482.1&id=8843816), this message can be ignored. [Bug 8843816](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=1443482.1&id=8843816) has been fixed in 11.1.  ***5. LB deadlock***  Global Wait-For-Graph(WFG) at ddTS[0.390] :  BLOCKED 0x3bc3e1b48 5 wq 2 cvtops x0 [0xd2703c03][0x545a14a5],[**LB**] [7B000-0002-00006C9D] 1  BLOCKER 0x3bc2efad0 5 wq 1 cvtops x0 [0xd2703c03][0x545a14a5],[**LB**] [48000-0001-000069FE] 0  BLOCKED 0x3bc2ed1c0 3 wq 2 cvtops x0 [0x415d0160][0xca28e8cf],[**LB**] [48000-0001-000069FE] 0  BLOCKER 0x3bc2dc648 3 wq 2 cvtops x0 [0x415d0160][0xca28e8cf],[**LB**] [34000-0001-0000095E] 0  BLOCKED 0x3bc2dc648 5 wq 2 cvtops x0 [0x415d0160][0xca28e8cf],[**LB**] [34000-0001-0000095E] 0  BLOCKER 0x3bc2dbbb0 5 wq 1 cvtops x0 [0x415d0160][0xca28e8cf],[**LB**] [76000-0002-000074D4] 1  BLOCKED 0x3bc2ef830 3 wq 2 cvtops x0 [0xd2703c03][0x545a14a5],[**LB**] [76000-0002-000074D4] 1  BLOCKER 0x3bc3e1b48 3 wq 2 cvtops x0 [0xd2703c03][0x545a14a5],[**LB**] [7B000-0002-00006C9D] 1  LB lock type refers to library cache lock. This type of deadlock is usually caused by a bug.  For example: [Bug 6475688](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&sourceId=1443482.1&id=6475688)  Concurrent rewrite and on-commit refresh can deadlock (library cache pin <--> lock) [Document 6475688.8](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=6475688.8)  The bug has been fixed in 11.1.0.7 and 11.2. Please apply patch accordingly.  ***6. Known Issues***  For other deadlock type or known issues related to dead lock, refer to [Document 554567.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1443482.1&id=554567.1) Summary Of Bugs Which Could Cause Deadlock In RAC Environment  ***7. Further Diagnosis***  Please collect the following information for further diagnosis:  a. alert log lmd0, and trace mentioned in the alert log from all instances.  b. set the following event to collect systemstate dump **ONLY** if the information in trace files are insufficient:  alter system set events '60 trace name systemstate level 258';  It will cause a systemstate dump to be generated whenever a deadlock is reported. If there are constant deadlocks, it could cause a lot of trace files being generated, monitor the system carefully.  To turn off the trace:  alter system set events '60 trace name context off'; |