



★ Troubleshooting Library Cache: Lock, Pin and Load Lock (Doc ID 444560.1)

[To Bottom](#)

In this Document

[Purpose](#)[Troubleshooting Steps](#)[What is "Library cache lock"?](#)[What is "Library cache pin"?](#)[Why are these two type of locks needed?](#)[How can Library Cache Contention be reduced in General?](#)[How can Library cache lock be reduced?](#)[How can Library cache pin be reduced?](#)[What is Library cache load lock?](#)[How can Library cache load lock be reduced?](#)[How are library cache pins and library load lock related?](#)[Hints](#)[Known Issues:](#)[References](#)

Was this document helpful?

☐ Yes
☐ No

Document Details



Type: TROUBLESHOOT
 Status: PUBLISHED
 Last Major Update: Aug 4, 2018
 Last Update: Mar 18, 2022
 Language: English

Related Products

Oracle Database Cloud Schema Service
 Oracle Database Exadata Express Cloud Service
 Gen 1 Exadata Cloud at Customer (Oracle Exadata Database Cloud Machine)
 Oracle Cloud Infrastructure - Database Service
 Oracle Database Backup Service

[Show More](#)

Information Centers

Oracle Catalog: Service Request Data Collections (SRDCs) for all Products and Services [51.2]

Oracle Catalog: Information Centers and Advisors for All Products and Services [50.2]

Information Center: OCI Cloud Guard [2765346.2]

Platform as a Service (PaaS) and Oracle Cloud Infrastructure (OCI) Information Center [2048297.2]

Index of Oracle Database Information Centers [1568043.2]

[Show More](#)

Document References

WAITEVENT: "cursor: pin S wait on X" Reference Note [1298015.1]

WAITEVENT: "library cache pin" Reference Note [34579.1]

Troubleshooting: Understanding and Tuning the Shared Pool [62143.1]

Bug 7423411 - Process may hang waiting for "library cache load lock" with no holder [7423411.8]

How to Find which Session is Holding a Particular Library Cache Lock [122793.1]

[Show More](#)

APPLIES TO:

Oracle Database Cloud Schema Service - Version N/A and later
 Oracle Database Exadata Express Cloud Service - Version N/A and later
 Oracle Database Exadata Cloud Machine - Version N/A and later
 Oracle Cloud Infrastructure - Database Service - Version N/A and later
 Oracle Database Backup Service - Version N/A and later
 Information in this document applies to any platform.

PURPOSE

This document provides brief information on Library cache lock, Library cache pin and Library cache load lock wait events.

TROUBLESHOOTING STEPS

What is "Library cache lock"?

This event controls the concurrency between clients of the library cache. It acquires a lock on the object handle so that either:

- One client can prevent other clients from accessing the same object.
- The client can maintain a dependency for a long time (for example, so that no other client can change the object).

This lock is also obtained to locate an object in the library cache.

Library cache lock will be obtained on database objects referenced during parsing or compiling of SQL or PL/SQL statements (table, view, procedure, function, package, package body, trigger, index, cluster, synonym). The lock will be released at the end of the parse or compilation.

Cursors (SQL and PL/SQL areas), pipes and any other transient objects do not use this lock.

Library cache lock is not deadlock sensitive and the operation is synchronous.

Parameters:

- handle address
Address of the object being loaded.
- lock address
Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is a State Object.
- Mode
Indicates the data pieces of the object which need to be loaded.

- Namespace
The name of the object namespace as it is displayed in V\$DB_OBJECT_CACHE view

What is "Library cache pin"?

This event manages library cache concurrency. Pinning an object causes the heaps to be loaded into memory. If a client wants to modify or examine the object, the client must acquire a pin after the lock. Pin can be acquired in NULL, SHARE or EXCLUSIVE modes and can be considered as a special form of lock. A wait for a "library cache pin" implies some other session holds that PIN in an incompatible mode.

Library cache pin will be obtained on a database object that is currently cached in the library cache (table, view, procedure, function, package, package body, trigger, index, cluster, synonym). In the library cache, a database object is cached in 2 parts: "handle" and "object". Library cache pin is only held when the "object" part is cached.

It is not deadlock sensitive and the operation is synchronous.

NOTE: In 10g and later, "library cache pins" are replaced by mutexes.

See: [Document 1298015.1](#) WAITEVENT: "cursor: pin S wait on X" Reference Note

Recently Viewed

[Resolving Issues For Library Cache Load Lock \[1477123.1\]](#)
[Cross-Platform Database Migration \(across same endian\) using RMAN backupset in 12c \[2013540.1\]](#)
[Cluster Interconnect in Oracle RAC \[787420.1\]](#)

Why are these two type of locks needed?

Both locks and pins are provided to access objects in the library cache. Locks manage concurrency between processes, whereas pins manage cache coherence. In order to access an object, a process must first lock the object handle, and then pin the object heap itself.

Requests for both locks and pins will wait until granted. This is a possible source of contention, because there is no NOWAIT request mode.

By acquiring a lock on the object handle, a process can prevent other processes from accessing the object, or even finding out what type it is. It can maintain a dependency on an object without preventing other processes from accessing the object. Acquiring a lock is also the only way to locate an object in cache. A process locates and locks an object in a single operation.

If the process wants to actually examine or modify the object, then it must acquire a pin on the object itself (after acquiring a lock on the handle). Pinning the object causes information about the object and its heaps to be loaded into memory if they were not already there. This information is guaranteed to remain in memory at least until the pin is released.

Oracle needs Library Cache Lock and Library Cache Pin for the compilation/parsing of Package/Procedure/Function/View. This is to ensure that no one is using the object during compilation/parsing while changes are being made to the definition of the object and the object dropped and recreate the object with the new definition.

When a SQL statement is hard parsed by a session, the session has to acquire a library cache lock to prevent other sessions from accessing or modifying the same object. If this event accumulates a lot of time, then it may indicate that the shared pool is too small or is being flushed regularly. Otherwise it indicates that database object definitions are being changed regularly.

Apart from hard parsing, if the session wants to change the definition of the object specified in the SQL or do any modifications, then it has to acquire a library cache lock along with the library cache pin. The pin is required because it needs the dictionary information to be loaded into memory to access the same code.

[Document 34579.1](#) WAITEVENT: "library cache pin" Reference Note

How can Library Cache Contention be reduced in General?

The following sections provide specific ways of addressing the various events that you may see as a result of contention. However, these waits or often a symptom of a general issue with cursor versions. If you see contention for any library cache events, it is worth immediately checking in an AWR report that you do not have hundreds or even thousands of versions of cursors:

SQL ordered by Version Count

• Only Statements with Version Count greater than 20 are displayed

Version Count	Executions	SQL Id	SQL Module	SQL Text
148,635		6n7hnnvnw7fcz8		
144,946		06v14z9s8jzc7		
1,019	230	95w0nq4tq9s7c		
979	310	94dhw6q5asj2r		
967	281	66w361y5df55r		

Any Version count > 20 is large and potentially detrimental to performance.

Version counts in to the thousands, such as those shown here, can be catastrophic.

if you do then using the following to resolve that:

[Document 296377.1](#) Troubleshooting: High Version Count Issues

How can Library cache lock be reduced?

We need to first determine whether the library cache contention is system-wide or related to a specific sql or set of sqls. Is the "library cache lock" being held for a long time by one specific query or for one specific object or is it being requested by many queries for a short duration and thereby causing high contention.

If the problem appears to be system-wide, high library cache locks is usually a result of an under-sized shared pool or non-sharing of sql. Some ways of reducing the contention are:

- Reduce the reloads by increasing the shared pool size as the locks may take a long time if the pool is undersized.
- Increase sharing by setting the cursor_sharing to similar or force.
Be aware this may change the execution plan; so setting the parameter should be thoroughly tested.
- Reduce the invalidations by running the batch jobs to collect statistics or any other maintenance jobs separately from OLTP.

The Oracle Performance Diagnostic Guide (OPDG) provides a number of extremely useful diagnostic scenarios for this wait:

wait: library cache lock

The library cache lock controls the concurrency between clients of the library cache by acquiring a lock on the object handle so that either:

- one client can prevent other clients from accessing the same object
- The client can maintain a dependency for a long time (no other client can change the object).

This lock is also obtained as part of the operation to locate an object in the library cache (a library cache child latch is obtained to scan a list of handles, then the lock is placed on the handle once the object has been found).

What to look for

- TKProf:
 - Overall wait event summary for non-recursive and recursive statements shows significant amount of time for *library cache lock* waits.
- AWR or statspack:
 - Significant waits for *library cache lock*

Cause Identified....

The Oracle Performance Diagnostic Guide (OPDG) can be downloaded from the following article:

[Document 390374.1](#) Oracle Performance Diagnostic Guide (OPDG)

If you find that a particular sql or set of sqls is causing the problem, it is necessary to determine the reason why the lock being held for a long time.

See the following Document in order to determine who is holding the 'Library Cache Lock' and on what object.

[Document 122793.1](#) How to Find which Session is Holding a Particular Library Cache Lock

How can Library cache pin be reduced?

If "library cache pin" waits form a significant amount of the wait time then it is important to determine if this is one or two sessions waiting long periods of time or a more serious general contention issue among lots of processes:

- If there is a blocking scenario where one or two processes are being blocked by one process, the reason for the process not releasing the pin needs to be determined
- If there is general widespread waiting then the shared pool may need tuning. See:

[Document 62143.1](#) Troubleshooting: Tuning the Shared Pool and Tuning Library Cache Latch Contention

What is Library cache load lock?

The session tries to find the load lock for the database object so that it can load the object.

The load lock is always obtained in exclusive mode, so that no other process can load the same object. If the load lock is busy the session will wait on this event until the lock becomes available.

Wait Time: 3 seconds (1 second for PMON)

Parameters:

- object address
Address of the object being loaded.
- lock address
Address of load lock being used

How can Library cache load lock be reduced?

If an object is not in memory, then a library cache lock cannot be acquired on it.

The object has to be loaded into the memory to acquire the lock.

The session tries to find the load lock for the database object so that it can load the object.

In order to prevent multiple processes requesting the load of the same object simultaneously, the other requesting sessions have to wait for the library cache load lock as the lock is busy with loading the object into the memory.

The waits on the library cache load lock is due to the objects not being available in memory.

The unavailability of the library cache object in the library cache is due to the undersized shared pool causing frequent reloads or too many hard parse as a result of unshared sqls.

To avoid this, the general recommendation would be:

- Increase the shared pool (to avoid high reloads)
- Increase the session cached cursors (to avoid the cursors flushing out of shared pool)
- Set cursor_sharing to force (to reduce hard parsing)--again this may change the plan and performance of query. This will change literals to use binds; so plan may change:

[Document 94036.1](#) Init.ora Parameter "CURSOR_SHARING" Reference Note

How are library cache pins and library load lock related?

Library cache pins and load locks may occur during PL/SQL, views, types, etc. compilation or recompilation. The compilation is always explicit (application installation, upgrades, applying patches),but object recompilation may happen transparently due to object invalidations.

Dealing with slow downs related to "mysterious" library cache pins and load locks we should look for the reason of the database object invalidations. They are likely to be triggered by actions causing changes to "LAST_DDL" attribute of database objects that have other dependent ones. Typically they are the object maintenance operations - ALTER, GRANT, REVOKE, replacing views, etc. Also gathering optimizer statistics will invalidate the cursor, causing reload in the library cache. This behavior is described in Oracle Server Application Developer's Guide as object dependency maintenance.

After object invalidation, Oracle tries to recompile the object at the time of the first access to it. It may be a problem when other sessions have pinned the object in the library cache. It is obvious that it is more likely to occur with more active users and with more complex dependencies (eg. many cross-dependent packages or package bodies). In some cases waiting for object recompilation may even take hours blocking all the sessions trying to access it.

This may be seen in library cache dump, level 10:

Look for ALTER ... COMPILE statement and objects/handles with lock=X or pin=X.

Hints

- Be very careful with altering, granting or revoking privileges on database objects that frequently used stored PL/SQL is dependent on. In fact, resolving this issue mostly depends on application project and system maintenance practices. Application developers should also consider that some project decisions have negative impact to the application scalability and performance.
- The load lock is obtained in EXCLUSIVE mode.
If the load lock is busy the session will wait on this event until the lock becomes available.

Known Issues:

[Document 1309738.1](#) High 'library cache lock' Wait Time Due to Invalid Login Attempts

[Document 10018789.8](#) Spin in kgllock / DB hang with high library cache lock waits

[Document 7423411.8](#) Process may hang waiting for "library cache load lock" with no holder

[Document 7706138.8](#) Process may hang waiting for "library cache load lock" with no holder

[Document 9675816.8](#) Bug 9675816 - Self deadlock with 'library cache lock' waits

REFERENCES

[NOTE:1298015.1](#) - WAITEVENT: "cursor: pin S wait on X" Reference Note

[NOTE:34579.1](#) - WAITEVENT: "library cache pin" Reference Note

[NOTE:62143.1](#) - Troubleshooting: Understanding and Tuning the Shared Pool

[NOTE:7423411.8](#) - Bug 7423411 - Process may hang waiting for "library cache load lock" with no holder

[NOTE:122793.1](#) - How to Find which Session is Holding a Particular Library Cache Lock

[NOTE:9675816.8](#) - Bug 9675816 - Self deadlock with 'library cache lock' waits / OERI:17059

[NOTE:10018789.8](#) - Bug 10018789 - Spin in kgllock / DB hang with high library cache lock waits on ADG

[NOTE:296377.1](#) - Troubleshooting: High Version Count Issues

[NOTE:19450314.8](#) - Bug 19450314 - Unnecessary compiled PL/SQL invalidations in 12c

[NOTE:7706138.8](#) - Bug 7706138 - Process may hang waiting for "library cache load lock" with no holder
[NOTE:94036.1](#) - Init.ora Parameter "CURSOR_SHARING" Reference Note

Didn't find what you are looking for?

Ask in Community...

Related

Products

- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Service > Oracle Database Cloud Schema Service
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Service > Oracle Database Exadata Express Cloud 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 Backup Service > Oracle Database Backup Service
- Oracle Cloud > Oracle Platform Cloud > Oracle Database Cloud Exadata Service > Oracle Database Cloud Exadata 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 > Locking Issues
- Oracle Database Products > Oracle Database Suite > Oracle Database > Oracle Database - Personal Edition > RDBMS > Database Level Performance Issues (not SQL Tuning)
- Oracle Database Products > Oracle Database Suite > Oracle Database > Oracle Database - Standard Edition > RDBMS > Database Level Performance Issues (not SQL Tuning)

Keywords

CACHE; INVALIDATION; LIBRARY CACHE; LIBRARY CACHE LOAD LOCK; LIBRARY CACHE LOCK; LIBRARY CACHE PIN; LOCK; TROUBLESHOOT

Errors

ORA-4061; ORA-4065; ORA-600 [17285]; ORA-6508

Translations

English Source

Chinese 简体中文

Japanese 日本語

Korean 한국어

[Back to Top](#)