C

Oracle Wait Events

This appendix describes Oracle wait events.

It includes these topics:

- Classes of Wait Events
- Descriptions of Common Wait Event Parameters
- Descriptions of Wait Events

Information about wait events is displayed in three dynamic performance views:

- `V$SESSION_WAIT` displays the events for which sessions have just completed waiting or are currently waiting.
- `V$SYSTEM_EVENT` displays the total number of times all the sessions have waited for the events in that view.
- `V$SESSION_EVENT` is similar to `V$SYSTEM_EVENT`, but displays all waits for each session.

> **See Also:**
>
> "V$SESSION_WAIT", "V$SYSTEM_EVENT", and "V$SESSION_EVENT"

Many of these wait events are tied to the internal implementation of Oracle and therefore are subject to change or deletion without notice. Application developers should be aware of this and write their code to tolerate missing or extra wait events.

The following SQL statement displays an alphabetical list of all Oracle wait events and the wait class to which they belong:

```
SQL> SELECT name, wait_class FROM V$EVENT_NAME ORDER BY name;
```

C.1 Classes of Wait Events

Every wait event belongs to a class of wait event.

The following list describes each of the wait classes.

Administrative

Waits resulting from DBA commands that cause users to wait (for example, an index rebuild)

Application

Waits resulting from user application code (for example, lock waits caused by row level locking or explicit lock commands)

Cluster

Waits related to Oracle Real Application Clusters resources (for example, global cache resources such as 'gc cr block busy')

Commit

This wait class only comprises one wait event - wait for redo log write confirmation after a commit (that is, 'log file sync')

Concurrency

Waits for internal database resources (for example, latches)

Configuration

Waits caused by inadequate configuration of database or instance resources (for example, undersized log file sizes, shared pool size)

Idle

Waits that signify the session is inactive, waiting for work (for example, 'SQL*Net message from client')

Network

Waits related to network messaging (for example, 'SQL*Net more data to dblink')

Other

Waits which should not typically occur on a system (for example, 'wait for EMON to spawn')

Queueing

Contains events that signify delays in obtaining additional data in a pipelined environment. The time spent on these wait events indicates inefficiency or other problems in the pipeline. It affects features such as parallel queries or `DBMS_PIPE` PL/SQL packages.

Scheduler

Resource Manager related waits (for example, 'resmgr: become active')

System I/O

Waits for background process I/O (for example, DBWR wait for 'db file parallel write')

User I/O

Waits for user I/O (for example 'db file sequential read')

C.2 Descriptions of Common Wait Event Parameters

This section describes some of the common wait event parameters.

block#

This is the block number of the block for which Oracle needs to wait. The block number is relative to the start of the file. To find the object to which this block belongs, issue the following SQL statement:

```
select segment_name, segment_type, owner, tablespace_name
from dba_extents
where file_id = file#
     and block#
     between block_id and block_id + blocks - 1;
```

blocks

The number of blocks that is being either read from or written to the file. The block size depends on the file type:

- Database files have a block size of `DB_BLOCK_SIZE`
- Logfiles and control files have a block size that is equivalent to the physical block size of the platform

break?

If the value for this parameter equals 0, a reset was sent to the client. A nonzero value indicates that a break was sent to the client.

class

The class of the block describes how the contents of the block are used. For example, class 1 represents data block, and class 2 represents sort block. Use this SQL query to see the block classes:

```
select view_definition
from v$fixed_view_definition
where view_name='GV$WAITSTAT';
```

dba

The initials "dba" represents the data block address, which consists of a file number and a block number.

driver id

The address of the disconnect function of the driver that is currently being used.

file#

The following query returns the name of the database file:

```
select *
from v$datafile
where file# = file#;
```

id1

The first identifier (*id1*) of the enqueue or global lock takes its value from P2 or P2RAW. The meaning of the identifier depends on the name (P1).

id2

The second identifier (*id2*) of the enqueue or global lock takes its value from P3 or P3RAW. The meaning of the identifier depends on the name (P1).

mode

The *mode* is usually stored in the low order bytes of P1 or P1RAW and indicates the mode of the enqueue or global lock request. This parameter has one of the following values:

Table C-1 Lock Mode Values

Mode Value	Description
1	Null mode
2	Sub-Share
3	Sub-Exclusive
4	Share
5	Share/Sub-Exclusive
6	Exclusive

Use the following SQL statement to retrieve the name of the lock and the mode of the lock request:

```
select chr(bitand(p1,-16777216)/16777215)||
       chr(bitand(p1, 16711680)/65535) "Lock",
       bitand(p1, 65535) "Mode"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

name and type

The name or "type" of the enqueue or global lock can be determined by looking at the two high order bytes of P1 or P1RAW. The name is always two characters. Use the following SQL statement to retrieve the lock name.

```
select chr(bitand(p1,-16777216)/16777215)||
      chr(bitand(p1,16711680)/65535) "Lock"
from v$session_wait
where event = 'DFS enqueue lock acquisition';
```

namespace

The name of the object namespace as it is displayed in `V$DB_OBJECT_CACHE` view.

requests

The number of I/Os that are "requested." This differs from the number of blocks in that one request could potentially contain multiple blocks.

session#

The number of the inactive session. Use the following SQL statement to find more information about the session:

```
select *
from v$session
where sid = session#;
```

waited

This is the total amount of time the session has waited for this session to terminate.

C.3 Descriptions of Wait Events

This section provides detailed descriptions for those wait events of greatest interest.

Where appropriate, pointers are provided to further information elsewhere in Oracle Database documentation. For a complete listing of wait events, in alphabetical order, you can issue the following SQL statement:

```
SQL> SELECT name FROM V$EVENT_NAME ORDER BY name;
```

C.3.1 alter system set dispatcher

A session has issued a statement `ALTER SYSTEM SET DISPATCHER = string` and is waiting for the dispatchers to get started.

Wait Time: The session will wait 1 / 100 of a second and verify if the new dispatchers have started, else the session will wait again

Parameter	Description
waited	Number of times that the session has waited 1 / 100 of a second

C.3.2 ARCH Remote Write

Used to track the time (in centiseconds) that ARC*n* background processes spend blocked waiting for network write operations to complete.

C.3.3 ASYNC Remote Write

Used to track the time (in centiseconds) for asynchronous streaming RFSWRITE operations. This includes stall reaps and streaming network submission time. This time is accumulated by TT*nn* (Redo Transport Worker) background processes.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.4 BFILE check if exists

The session waits to check if an external large object (LOB) exists.

Wait Time: The total elapsed time for the **exists** call

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.5 BFILE check if open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the **isopen** call

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.6 BFILE closure

The session waits for an external large object (LOB) to close.

Wait Time: The total elapsed time for the **close** call

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.7 BFILE get length

The session waits on a call to check the size of an external large object (LOB).

Wait Time: The total elapsed time for the call to check the LOB size

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.8 BFILE get name object

The session waits on a call to find or generate the external name of a external large object.

Wait Time: The total elapsed time for **make external file name** to complete

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.9 BFILE get path object

The session is waiting on a call to find or generate the external path name of an external large object (LOB).

Wait Time: The total elapsed time for **make external path** to complete

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.10 BFILE internal seek

The session waits for a positioning call within the external large object (LOB) to complete.

Wait Time: The total elapsed time for the **seek** to complete

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.11 BFILE open

The session waits for an external large object (LOB) to open.

Wait Time: The total elapsed time for the **isopen** call

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.12 BFILE read

The session waits for a read from a external large object (LOB) to complete.

Wait Time: The total elapsed time for the **read** to complete

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.13 broadcast mesg queue transition

Processes enter "wait for broadcast mesg queue transition" when cleaning up a publisher channel handle to a RELIABLE broadcast channel. The publisher is responsible for moving the message to the free queue, but it cannot do so until the message is in the done queue. If the message is still not in the done queue, process enters this wait. This wait event will most likely appear when an Oracle process is about to exit normally, or when PMON cleans up a terminated process.

Wait Time: Varies

Parameter	Description
channel handle	publisher channel handle pointer

Parameter	Description
message	broadcast message pointer
location	A number indicating the function in KSR where the process is waiting

C.3.14 broadcast mesg recovery queue transition

Processes enter "wait for broadcast mesg recovery queue transition" when cleaning up a publisher channel handle to a RELIABLE broadcast channel. The broadcasted message is in the recovery queue of another channel handle (for example, ch2). Process enters this wait, if the message is yet to be removed from the recovery queue of the ch2 channel handle. This wait event will most likely appear when an Oracle process is about to exit normally, or when PMON cleans up a terminated process.

Wait Time: Varies

Parameter	Description
channel handle	Publisher channel handle pointer
message	Broadcast message pointer
location	A number indicating the function in KSR where the process is waiting

C.3.15 buffer busy waits

Wait until a buffer becomes available.

There are four reasons that a session cannot pin a buffer in the buffer cache, and a separate wait event exists for each reason:

1. "buffer busy waits": A session cannot pin the buffer in the buffer cache because another session has the buffer pinned.
2. "read by other session": A session cannot pin the buffer in the buffer cache because another session is reading the buffer from disk.
3. "gc buffer busy acquire": A session cannot pin the buffer in the buffer cache because another session is reading the buffer from the cache of another instance.
4. "gc buffer busy release": A session cannot pin the buffer in the buffer cache because another session on another instance is taking the buffer from this cache into its own cache so it can pin it.

Prior to release 10.1, all four reasons were covered by "buffer busy waits." In release 10.1, the "gc buffer busy" wait event covered both the "gc buffer busy acquire" and "gc buffer busy release" wait events.

Wait Time: Normal wait time is 1 second. If the session was waiting for a buffer during the last wait, then the next wait will be 3 seconds.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about the potential causes of the buffer busy waits wait event

C.3.16 buffer deadlock

Oracle does not really wait on this event; the foreground only yields the CPU. Thus, the chances of catching this event are very low. This is not an application induced deadlock, but an assumed deadlock by the cache layer. The cache layer cannot get a buffer in a certain mode within a certain amount of time.

Wait Time: 0 seconds. The foreground process only yields the CPU and will usually be placed at the end of the CPU run queue.

Parameter	Description
class	See "class"
mode	See "mode"
flag	The flag points to the internal flags used by the session to get this block
dba	See "dba"

C.3.17 buffer latch

The session waits on the buffer hash chain latch. Primarily used in the dump routines.

Wait Time: 1 second

Parameter	Description

latch addr	The virtual address in the SGA where this latch is located. Use the following statement to find the name of this latch:
	`select *`
	`from v$latch a, v$latchname b`
	`where addr = latch addr`
	`and a.latch# = b.latch#;`

chain#	The index into array of buffer hash chains. When the chain is 0xffffffff, the foreground waits on the LRU latch.

C.3.18 buffer read retry

This event occurs only if the instance is mounted in shared mode (Oracle Real Application Cluster).
During the read of the buffer, the contents changed. This means that either:

- The version number, dba, or the incarnation and sequence number stored in the block no longer match
- The checksum on the block does not match the checksum in the block

The block will be reread (this may fail up to 3 times), then corruption is assumed and the corrupt block is dumped in the trace file.

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
file#	See "file#"
block#	See "block#"

C.3.19 checkpoint completed

A session waits for a checkpoint to complete. This could happen, for example, during a close database or a local checkpoint.

Wait Time: 5 seconds

Parameters: None

C.3.20 cleanup of aborted process

When a process spawn is terminated while the process spawning background is in the middle of spawning, the current session must wait until the pid of the new process is filled in. Once the pid is filled in, then the process spawn can be actually terminated.

Wait Time: Usually 3 seconds

Parameter	Description
location	Location of the wait

C.3.21 control file parallel write

This event occurs while the session is writing physical blocks to all control files. This happens when:

- The session starts a control file transaction (to ensure the control files are up to date in case the session crashes before committing the control file transaction)
- The session commits a transaction to a control file
- Changing a generic entry in the control file, the new value is being written to all control files

Wait Time: The wait time is the time it takes to finish all writes to all control files

Parameter	Description
files	The number of control files to which the session is writing
blocks	The number of blocks that the session is writing to the control file
requests	The number of I/O requests which the session wants to write

C.3.22 control file sequential read

Reading from the control file. This happens often. For example, while:

- Making a backup of the control files
- Sharing information (between instances) from the control file
- Reading other blocks from the control files

- Reading the header block

Wait Time: The wait time is the elapsed time of the read

Parameter	Description
file#	The control file from which the session is reading
block#	Block number in the control file from where the session starts to read. The block size is the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to read

C.3.23 control file single write

This wait is signaled while the control file's shared information is written to disk. This is an atomic operation protected by an enqueue (CF), so that only one session at a time can write to the entire database.

Wait Time: The wait time is the elapsed time of the write

Parameter	Description
file#	This identifies the control file to which the session is currently writing
block#	Block number in the control file where the write begins. The block size is the as the physical block size of the port (usually 512 bytes, some UNIX ports have 1 or 2 Kilobytes).
blocks	The number of blocks that the session is trying to write

C.3.24 cursor: mutex S

A session waits on this event when it is requesting a mutex in shared mode, when another session is currently holding a this mutex in exclusive mode on the same cursor object.

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contain SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

C.3.25 cursor: mutex X

The session requests the mutex for a cursor object in exclusive mode, and it must wait because the resource is busy. The mutex is busy because either the mutex is being held in exclusive mode by another session or the mutex is being held shared by one or more sessions. The existing mutex holder(s) must release the mutex before the mutex can be granted exclusively.

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contain SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)

Parameter	Description
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

C.3.26 cursor: pin S

A session waits on this event when it wants to update a shared mutex pin and another session is currently in the process of updating a shared mutex pin for the same cursor object. This wait event should rarely be seen because a shared mutex pin update is very fast.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

C.3.27 cursor: pin S wait on X

A session waits for this event when it is requesting a shared mutex pin and another session is holding an exclusive mutex pin on the same cursor object.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

C.3.28 cursor: pin X

A session waits on this event when it is requesting an exclusive mutex pin for a cursor object and it must wait because the resource is busy. The mutex pin for a cursor object can be busy either because a session is already holding it exclusive, or there are one or more sessions which are holding shared mutex pin(s). The exclusive waiter must wait until all holders of the pin for that cursor object have released it, before it can be granted.

Wait Time: Microseconds

Parameter	Description
P1	Hash value of cursor
P2	Mutex value (top 2 bytes contains SID holding mutex in exclusive mode, and bottom two bytes usually hold the value 0)
P3	Mutex where (an internal code locator) OR'd with Mutex Sleeps

C.3.29 db file async I/O submit

When asynchronous I/O is available, this wait event captures any time spent in submitting I/Os to the underlying storage.

> **See Also:**
>
> "db file parallel write"

C.3.30 db file parallel read

This happens during recovery. It can also happen during buffer prefetching, as an optimization (rather than performing multiple single-block reads). Database blocks that must be changed as part of recovery are read in parallel from the database.

Wait Time: Wait until all of the I/Os are completed

Parameter	Description
files	This indicates the number of files to which the session is reading
blocks	This indicates the total number of blocks to be read
requests	This indicates the total number of I/O requests, which will be the same as blocks

C.3.31 db file parallel write

This event occurs in the DBWR. It indicates the time that DBWR spends waiting for I/O completion.

If asynchronous I/O is available, then the db file async I/O submit wait event captures any time spent in submitting I/Os to the underlying storage.

When asynchronous I/O is not available, db file parallel write captures the time spent during submit and reap.

Wait Time: While there are outstanding I/Os, DBWR waits for some of the writes to complete. DBWR does not wait for all of the outstanding I/Os to complete.

Parameter	Description
requests	This indicates the total number of I/O requests, which will be the same as blocks
interrupt	
timeout	This indicates the timeout value in hundredths of a second to wait for the I/O completion.

See Also:

- *Oracle Database Performance Tuning Guide* about how this wait event can help identify I/O problems
- "db file async I/O submit"

C.3.32 db file scattered read

Similar to the “db file sequential read” wait event, except that the session is reading multiple data blocks.

Wait Time: The wait time is the actual time it takes to do all of the I/Os

Parameter	Description
file#	See "file#"
block#	See "block#"
blocks	The number of blocks that the session is trying to read from the *file#* starting at *block#*

See Also:

- *Oracle Database Performance Tuning Guide* for more information about this wait event
- *Oracle Database Performance Tuning Guide* about how this wait event can help identify I/O problems
- *Oracle Database Performance Tuning Guide* about potential causes for this wait event

C.3.33 db file sequential read

The session waits while a sequential read from the database is performed. This event is also used for rebuilding the control file, dumping data file headers, and getting the database file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
file#	See "file#"
block#	See "block#"
blocks	This is the number of blocks that the session is trying to read (should be 1)

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for more information about this wait event
> - *Oracle Database Performance Tuning Guide* for more information about how this wait event can help identify I/O problems
> - *Oracle Database Performance Tuning Guide* about potential causes for this wait event

C.3.34 db file single write

This event is used to wait for the writing of the file headers.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
file#	See "file#"
block#	See "block#"
blocks	This is the number of blocks that the session is trying to write in *file# starting at block#*

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* about how this wait event can help identify I/O problems

C.3.35 DFS lock handle

The session waits for the lock handle of a global lock request. The lock handle identifies a global lock. With this lock handle, other operations can be performed on this global lock (to identify the global lock in future operations such as conversions or release). The global lock is maintained by the DLM.

Wait Time: The session waits in a loop until it has obtained the lock handle from the DLM. Inside the loop there is a wait of 0.5 seconds.

Parameter	Description
name	See "name and type"
mode	See "mode"
id1	See "id1"
id2	See "id2"

C.3.36 direct path read

During Direct Path operations the data is asynchronously read from the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if during a direct read no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	Number of valid buffers in the context referenced by the descriptor address

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.37 direct path sync

During Direct Path write operations the data is asynchronously written to the database files. At some point the session needs to ensure that all outstanding asynchronous I/O have been completed to disk. On UNIX the `fsync` command, which synchronizes data to disk, is issued to confirm that all the writes have completed and the data is all on disk.

Wait Time: The time taken for the fsync operation to complete, which normally is the time taken to complete the outstanding I/Os.

Parameters	Description
file#	See "file#"
flags	Flags used for debugging purposes

C.3.38 direct path write

During Direct Path operations, the data is asynchronously written to the database files. At some stage the session needs to make sure that all outstanding asynchronous I/O have been completed to disk. This can also happen if, during a direct write, no more slots are available to store outstanding load requests (a load request could consist of multiple I/Os).

Wait Time: 10 seconds. The session will be posted by the completing asynchronous I/O. It will never wait the entire 10 seconds. The session waits in a tight loop until all outstanding I/Os have completed.

Parameter	Description
descriptor address	This is a pointer to the I/O context of outstanding direct I/Os on which the session is currently waiting
first dba	The dba of the oldest I/O in the context referenced by the descriptor address
block cnt	Number of valid buffers in the context referenced by the descriptor address

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this parameter

C.3.39 Disk file operations I/O

This event is used to wait for disk file operations (for example, open, close, seek, and resize). It is also used for miscellaneous I/O operations such as block dumps and password file accesses.

Wait Time: The wait time is the actual time it takes to do the I/O

Parameter	Description
FileOperation	Type of file operation
fileno	File identification number
filetype	Type of file (for example, log file, data file, and so on)

C.3.40 dispatcher shutdown

During shutdown immediate or normal, the shutdown process must wait for all the dispatchers to shutdown. As each dispatcher is signaled, the session that causes the shutdown waits on this event until the requested dispatcher is no longer running.

Wait Time: 1 second

Parameter	Description
waited	Indicates the cumulative wait time. After 5 minutes, the session writes to the alert and trace files to indicate that there might be a problem.

> **See Also:**
>
> *Oracle Database SQL Language Reference* for information about shutting down a dispatcher using `SHUTDOWN` clause of the SQL `ALTER SYSTEM` statement

C.3.41 dispatcher timer

This basically means that the dispatcher is idle and waiting for some work to arrive.

Wait Time: 60 seconds

Parameter	Description
sleep time	The intended sleep time. The dispatcher will return to work sooner if it is posted by either data arriving on the network or by a post from a shared server process to send data back to the client.

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.42 dupl. cluster key

It is possible for a race condition to occur when creating a new cluster key. If it is found that another process has put the cluster key into the data/index block, then the session waits and retries. The retry should then find a valid cluster key.

Wait Time: 0.01 seconds

Parameter	Description
dba	The dba of the block into which the session is trying to insert a cluster key

C.3.43 enq: IQ - Text index maintenance

Used for coordinating concurrent execution of Oracle Text index maintenance events. Serialization behavior depends on the event type.

Wait Time: Varies. Depends on the duration of the Oracle Text index maintenance event type.

Parameter	Description
event type	Event type To interpret this numeric value, refer to the view "V$TEXT_WAITING_EVENTS". The list of values for the `EVENT_TYPE` column is ordered by event type number, starting with `0`. For example, event type `0` = `NONE`, event type `1` = `SYNC-Mapping Timeout`, event type `2` = `SYNC-Mapping`, and so on.
objn	Object number of the Oracle Text index or index partition

C.3.44 enq: OW - initialization

A session will wait on this event if it is trying to initialize the database wallet, and another session has already begun an initialization.

Wait Time: Total time necessary to initialize the wallet context

Parameters: None

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about the potential causes of an enqueue wait event

C.3.45 enq: OW - termination

A session will wait on this event if it is trying to terminate the database wallet, and another session has already begun a termination.

Wait Time: Total time necessary to deallocate memory used by the wallet context and terminate the context.

Parameters: None

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about the potential causes of an enqueue wait event

C.3.46 enq: SV - contention

A session is accessing a sequence number from an ordered sequence object.

Wait Time: The wait time is the actual time waited until the requested sequence number is acquired.

Parameter	Description
id1	Sequence object#

> **See Also:**
>
> *Oracle Database SQL Language Reference* for more information about sequences

C.3.47 enq: TX - index contention

Occurs when a transaction inserting a row in an index has to wait for the end of an index block split being done by another transaction.

C.3.48 enq: TX - row lock (HIGH priority)

This wait event occurs when a transaction with `HIGH` priority is waiting for a row lock being held by another transaction.

This wait event is useful when configuring Priority Transactions. You can observe the statistics for this wait event to determine whether the wait time for `HIGH` priority transactions is appropriate for your environment.

Wait Time: The amount of time specified by the `PRIORITY_TXNS_HIGH_WAIT_TARGET` initialization parameter.

Parameter	Description
name	See "name and type"
mode	See "mode"
usn<<16	Undo segment number
slot	Slot number
sequence	Sequence number

See Also:

- *Oracle Database Administrator's Guide* for information about configuring Priority Transactions
- "PRIORITY_TXNS_HIGH_WAIT_TARGET"

C.3.49 enq: TX - row lock (LOW priority)

This wait event occurs when a transaction with `LOW` priority is waiting for a row lock being held by another transaction.

This wait event is useful when configuring Priority Transactions. You can observe the statistics for this wait event to determine how long `LOW` priority transactions are waiting for row locks.

Wait Time: Total time necessary to obtain the row lock

Parameter	Description
name	See "name and type"
mode	See "mode"
usn<<16	Undo segment number
slot	Slot number
sequence	Sequence number

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for information about configuring Priority Transactions

C.3.50 enq: TX - row lock (MEDIUM priority)

This wait event occurs when a transaction with `MEDIUM` priority is waiting for a row lock being held by another transaction.

This wait event is useful when configuring Priority Transactions. You can observe the statistics for this wait event to determine whether the wait time for `MEDIUM` priority transactions is appropriate for your environment.

Wait Time: The amount of time specified by the `PRIORITY_TXNS_MEDIUM_WAIT_TARGET` initialization parameter.

Parameter	Description
name	See "name and type"
mode	See "mode"
usn<<16	Undo segment number
slot	Slot number
sequence	Sequence number

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for information about configuring Priority Transactions
> - "PRIORITY_TXNS_MEDIUM_WAIT_TARGET"

C.3.51 enq: TX - row lock contention

This wait event can occur for several reasons.

- If one user is wanting to update or delete a row or rows that another session is modifying. The session holding the lock will release it when it performs a COMMIT or ROLLBACK.
- If a session is waiting due to potential duplicates in a UNIQUE index. If two sessions try to insert the same key value, the second session has to wait to see if an ORA-0001 should be raised or not. The session holding the lock will release it when it performs a COMMIT or ROLLBACK.
- If a session is waiting due to a shared bitmap index fragment. Bitmap indexes index key values and a range of rowids. Each entry in a bitmap index can cover many rows in the actual table. If two sessions want to update rows covered by the same bitmap index fragment, then the second session waits for the first transaction to either COMMIT or ROLLBACK by waiting for the TX lock.

C.3.52 enqueue

The session is waiting for a local enqueue.

The wait depends on the name of the enqueue (see Oracle Enqueue Names).

Wait Time: Depends on the enqueue name

Parameter	Description
name	See "name and type"
mode	See "mode"

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about the potential causes of an enqueue wait event

C.3.53 flashback buf free by RVWR

This wait event only occurs when Flashback Database is turned on. A session waits for recovery writer (RVWR) to write flashback data to the flashback logs on disk because the buffers are full. Until RVWR can free up the buffers, the session may need to wait.

If this event becomes a top wait event for the database, it is typically because the file system or storage system for the Fast Recovery Area does not support enough bandwidth for Oracle to write the flashback database logs. Refer to the Flashback Database section in *Oracle Database Backup and Recovery User's Guide* for tuning considerations.

Wait Time: 1 second

Parameters: None

C.3.54 flashback log file sync

Waits for flashback database data to be written to disk.

Wait Time: Includes RVWR writing the flashback database data and posting this process

Parameters: None

C.3.55 free buffer waits

This wait event can occur for several reasons.

- All buffer gets have been suspended. This could happen when a file was read-only and is now read/write. All the existing buffers must be invalidated since they are not linked to lock elements (needed when mounted parallel (shared)). So cache buffers are not assigned to data block addresses until the invalidation is finished.
- The session moved some dirty buffers to the dirty queue and now this dirty queue is full. The dirty queue must written first. The session will wait on this event and try again to find a free buffer

- This also happens after inspecting **free buffer inspected** buffers. If no free buffer is found, Oracle waits for one second, and then tries to get the buffer again (depends on the context). For more information, see free buffer inspected.

Wait Time: 1 second

Parameter	Description
file#	See "file#"
block#	See "block#"

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for more information about this wait event
> - *Oracle Database Performance Tuning Guide* about potential causes of this wait event

C.3.56 free global transaction table entry

The session is waiting for a free slot in the global transaction table (used by the Distributed Database option). It will wait for 1 second and try again.

Wait Time: 1 second

Parameter	Description
tries	The number of times the session tried to find a free slot in the global transaction table

C.3.57 free process state object

Used during the creation of a process. The session will scan the process table and look for a free process slot. If none can be found, PMON is posted to check if all the processes currently in the process table are still running. If there are terminated processes, then PMON will clean them and make the process slot available to new processes. The waiting process will then rescan the process table to find the new slot.

Wait Time: 1 second

Parameters: None

C.3.58 gc buffer busy acquire

This event indicates that the requested buffer was globally busy in the cluster, because it had already been requested from a remote instance by another local client.

Wait Time: The wait time is the actual time it took to resolve the buffer contention.

Parameter	Description
file#	See "file#"

Parameter	Description
block#	See "block#"
class#	See "class"

C.3.59 gc buffer busy release

This event indicates that the requested buffer was globally busy in the cluster, because the requested buffer had been transferred away by a remote instance and needed to be requested again.

Wait Time: The wait time is the actual time it took to resolve the buffer contention.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.60 gc cr block 2-way

This event indicates that the requested consistent read (CR) buffer was transferred from another instance in the cluster, and 2 network hops were involved in the request. The Cache Fusion protocol guarantees that requests are fulfilled in 3 hops or less, regardless of the number of instances in the cluster.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.61 gc cr block 3-way

This event indicates that the requested consistent read (CR) buffer was transferred from another instance in the cluster, and 3 network hops were involved in the request. The Cache Fusion protocol guarantees that requests are fulfilled in 3 hops or less, regardless of the number of instances in the cluster.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.62 gc cr block busy

This event indicates that the requested consistent read (CR) buffer was transferred from another instance in the cluster, but the request had to be put on hold at the serving instance.

This could happen when a dirty buffer had to be transferred, in which case REDO needs to be flushed before the Cache Fusion transfer could be processed. This could also happen if the requested buffer was being held by another process on the remote instance. The serving instance would transfer the requested buffer once the REDO flush is complete, or once the buffer is released by the holding process.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.63 gc cr block congested

This event indicates that the requested consistent read (CR) buffer was transferred from another instance in the cluster, but the request was queued at the serving instance for too long.

This could happen when the GCS server processes (LMS*n* and LM*nn*) are very busy due to having a high volume of incoming Cache Fusion requests. Increasing the GCS server processes (`GCS_SERVER_PROCESSES`) is usually the remedy to alleviate congestion for the GCS server processes.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

> **See Also:**
>
> "GCS_SERVER_PROCESSES"

C.3.64 gc cr block direct read

This event indicates that the requested consistent read (CR) buffer was transferred from another instance in the cluster, and the buffer was directly read from the remote cache utilizing the RDMA protocol.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.65 gc cr block lost

This event indicates that a Cache Fusion consistent read (CR) buffer request resulted in a potentially lost block. The request will be re-tried automatically.

This is typically caused by unreliable interconnect network protocols, usually when the network is congested and packets are dropped. It can also be the result of interconnect hardware errors or simply the result of an overloaded system, or a sign that a public network path is used for Cache Fusion instead of a private network.

Wait Time: The wait time is the actual time it took to detect the potentially lost block.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.66 gc cr grant read-mostly invalidation

The client is performing a DML on a read-mostly object, which requires a read lock invalidation on all other instances.

Wait Time: The wait time is the actual time it takes to complete the invalidation on all other instances.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.67 gc cr multi block grant

This event indicates that a client requested multiple consistent read (CR) buffers from another instance in the cluster, and the request resulted in multiple lock grants from the remote instance.

Wait Time: The wait time is the actual time waited for the multi-block request to complete.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.68 gc cr multi block mixed

This event indicates that a client requested multiple consistent read (CR) buffers from another instance in the cluster, and the remote instance returned a mixture of blocks and grants.

Wait Time: The wait time is the actual time waited for the multi-block request to complete.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.69 gc cr request

This event is used for ongoing Cache Fusion consistent read (CR) buffer requests. The event is renamed to reflect the actual outcome of the request (example: "gc cr block 2-way") upon completion of the wait. Therefore, these events will not be reported in AWR reports, but may appear in ASH data.

Wait Time: The wait time is the actual time waited for the request.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.70 gc current block 2-way

This event indicates that the requested current buffer was transferred from another instance in the cluster, and 2 network hops were involved in the request. The Cache Fusion protocol guarantees that requests are fulfilled in 3 hops or less, regardless of the number of instances in the cluster.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

C.3.71 gc current block 3-way

This event indicates that the requested current buffer was transferred from another instance in the cluster, and 3 network hops were involved in the request. The Cache Fusion protocol guarantees that requests are fulfilled in 3 hops or less, regardless of the number of instances in the cluster.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

C.3.72 gc current block busy

This event indicates that the requested current buffer was transferred from another instance in the cluster, but the request had to be put on hold at the serving instance.

This could happen when a dirty buffer had to be transferred, in which case REDO needs to be flushed before the Cache Fusion transfer could be processed. This could also happen if the requested buffer was being held by another process on the remote instance. The serving instance would transfer the requested buffer once the REDO flush is complete, or once the buffer is released by the holding process.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

C.3.73 gc current block congested

This event indicates that the requested current buffer was transferred from another instance in the cluster, but the request was queued at the serving instance for too long.

This could happen when the GCS server processes (LMS) are very busy due to having a high volume of incoming Cache Fusion requests. Increasing the GCS server processes (`GCS_SERVER_PROCESSES`) is usually the remedy to alleviate congestion for the GCS server processes.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

> **See Also:**
>
> "GCS_SERVER_PROCESSES"

C.3.74 gc current block direct read

This event indicates that the requested current buffer was transferred from another instance in the cluster, and the buffer was directly read from the remote cache utilizing the RDMA protocol.

Wait Time: The wait time is the actual time it took to receive the requested buffer.

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.75 gc current block lost

This event indicates that a Cache Fusion current buffer request resulted in a potentially lost block. The request will be re-tried automatically.

This is typically caused by unreliable interconnect network protocols, usually when the network is congested and packets are dropped. It can also be the result of interconnect hardware errors or simply the result of an overloaded system, or a sign that a public network path is used for Cache Fusion instead of a private network.

Wait Time: The wait time is the actual time it took to detect the potentially lost block.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

C.3.76 gc current request

This event is used for ongoing Cache Fusion current buffer requests. The event is renamed to reflect the actual outcome of the request (example: "gc current block 2-way") upon completion of the wait. Therefore, these events will not be reported in AWR reports, but may appear in ASH data.

Wait Time: The wait time is the actual time waited for the request.

Parameter	Description
file#	See "file#"
block#	See "block#"
id#	Block class and global access mode held and requested

C.3.77 inactive session

This event is used for switching and terminating sessions.

- Switching sessions

If a timeout period has been specified, then wait that amount of time for the session to be detached.

- Terminating sessions

  From either `KILL SESSION` or internal request. Having posted a session that it should terminate itself, wait for up to 1 minute for the session to terminate.

Wait Time: 1 second

Parameter	Description
session#	See "session#"
waited	See "waited"

C.3.78 inactive transaction branch

The session waits for a transaction branch that is currently used by another session.

Wait Time: 1 second

Parameter	Description
branch#	The serial number of the transaction for which the session is waiting
waited	See "waited"

C.3.79 index split completion

This event occurs when a transaction inserting a row into an index must wait for the end of an index block split being performed by another transaction. In an Oracle RAC database, some index split waits will use this wait event in lieu of the "enq: TX - index contention" wait event.

Wait Time: The wait time is the actual time waited for the split to complete.

Parameter	Description
undo seg#	The rollback segment ID
slot#	The slot ID inside the rollback segment
wrap#	The sequence number that is incremented for each transaction
count	The number of times that the session has waited on this transaction

C.3.80 instance state change

The session waits for SMON to enable or disable cache or transaction recovery. This usually happens during `ALTER DATABASE OPEN` or `CLOSE`.

Wait Time: Wait time depends on the amount of time the action takes (that is, the amount of recovery needed)

Parameter	Description
layer	This value can be 1 or 2. If 1, it means that the transaction layer wants transaction recovery to be performed. If 2, it means that cache recovery will be performed.
value	This value can be 0 (disable) or 1 (enable)

Parameter	Description
waited	The number of seconds waited so far

C.3.81 io done

The session waits for an I/O to complete or it waits for a worker process to become available to submit the I/O request. This event occurs on platforms that do not support asynchronous I/O.

Wait Time: 50 milliseconds

Parameter	Description
msg ptr	A pointer to the I/O request

C.3.82 ksxr wait for mount shared

The cross instance broadcast facility of this Oracle instance is waiting for the database mount in shared mode to complete.

Wait Time: The time taken for the instance to mount. An indefinite wait on this event implies that the instance startup is hanging.

C.3.83 ktm: instance recovery

The session waits for SMON to finish the instance, transaction recovery, or sort segment cleanup.

Wait Time: The wait time can vary and depends on the amount of recovery needed

Parameter	Description

undo segment#	If the value is 0, SMON is probably performing instance recovery. If P1 > 0, use this query to find the undo segment:
	`select *`
	`from v$rollstat`
	`where usn = undo segment#;`

C.3.84 latch activity

This event is used as part of the process of determining whether a latch must be cleaned.

Wait Time: 0.05 to 0.1 seconds

Parameter	Description
address	The address of the latch that is being checked

number	The latch number of the latch that has activity. To find more information on the latch, use the following SQL statement:
	`select *`
	`from v$latchname`
	`where latch# = number;`

Parameter	Description
process#	If this is 0, it is the first phase of the in-flux tests

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about latch wait events

C.3.85 latch free

The process waits for a latch that is currently busy (held by another process).

Wait Time: The wait time increases exponentially and does not include spinning on the latch (active waiting). The maximum wait time also depends on the number of latches that the process is holding. There is an incremental wait of up to 2 seconds.

Parameter	Description
address	The address of the latch for which the process is waiting

number	The latch number that indexes in the `V$LATCHNAME` view.To find more information on the latch, use the following SQL statement:
	`select *`
	`from v$latchname`
	`where latch# = number;`

tries	A count of the number of times the process tried to get the latch (slow with spinning) and the process has to sleep

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about latch wait events

C.3.86 latch: redo copy

Redo copy latches are acquired by sessions in order to write changes into the redo log buffer and by the LGWR process to ensure that there are no current writers before it copies data from the buffer to the redo log.

C.3.87 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)

Parameter	Description
object address	Address of the object being loaded
lock address	Address of load lock being used
mask	Indicates which data pieces of the object that must loaded

C.3.88 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, no other client can change the object)

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

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
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 must be loaded
namespace	See "namespace"

See Also:

- *Oracle Database Performance Tuning Guide* for more information about this wait event
- *Oracle Database Performance Tuning Guide* for more information about potential causes of this wait event

C.3.89 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.

Wait Time: 3 seconds (1 second for PMON)

Parameter	Description
handle address	Address of the object being loaded
pin address	Address of the load lock being used. This is not the same thing as a latch or an enqueue, it is basically a State Object.
mode	Indicates which data pieces of the object that must be loaded
namespace	See "namespace"

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for more information about this wait event
> - *Oracle Database Performance Tuning Guide* for more information about potential causes of this wait event

C.3.90 library cache shutdown

The process shutting down the instance waits for sessions to complete before proceeding with library cache shutdown.

C.3.91 library cache: mutex X

Library cache mutexes protect hash buckets within the shared pool. Each hash bucket contains a number of cursors. The mutex must be held in exclusive mode before any of the structures that it protects can be changed. This wait event is often associated with high parse activity.

C.3.92 Log archive I/O

Used local archiving of online redo logs (for a production database) or standby redo logs (for a standby database). When the archiving process exhausts its I/O buffers because all of them are being used for on-going I/O's, the wait for an available I/O buffer is captured in this system wait event.

Wait Time: Depends on the speed of the disks

Parameters: None

C.3.93 log buffer space

Waiting for space in the log buffer because the session is writing data into the log buffer faster than LGWR can write it out. Consider making the log buffer bigger if it is small, or moving the log files to faster disks such as striped disks.

Wait Time: Usually 1 second, but 5 seconds if it is waiting for a Switch Logfile to complete

Parameters: None

> **See Also:**
>
> - *Oracle Database Performance Tuning Guide* for more information about this wait event
> - *Oracle Database Performance Tuning Guide* for more information about potential causes of this wait event

C.3.94 log file parallel write

Writing redo records to the redo log files from the log buffer.

Wait Time: Time it takes for the I/Os to complete. Even though redo records are written in parallel, the parallel write is not complete until the last I/O is on disk.

Parameter	Description
files	Number of files to be written
blocks	Number of blocks to be written
requests	Number of I/O requests

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for information about how this wait event can help identify I/O problems

C.3.95 log file sequential read

Waiting for the read from this logfile to return. This is used to read redo records from the log file.

Wait Time: Time it takes to complete the physical I/O (read)

Parameter	Description
log#	The relative sequence number of the logfiles within a log group (used only when dumping the logfiles)
block#	See "block#"
blocks	The number of blocks to read

C.3.96 log file single write

Waiting for the write to this logfile to complete. This event is used while updating the header of the logfile. It is signaled when adding a log file member and when incrementing sequence numbers.

Wait Time: Time it takes for the physical I/O (write) to complete

Parameter	Description
log#	This is the number of the group/log to which the session is currently writing
block#	See "block#"
blocks	The number of blocks to write

C.3.97 log file switch (archiving needed)

Waiting for a log switch because the log that the LGWR will be switching into has not been archived yet. Check the alert log to ensure that archiving has not stopped due to a failed archive write. To speed archiving, consider adding more archive processes or putting the archive files on striped disks.

Wait Time: 1 second

Parameters: None

C.3.98 log file switch (checkpoint incomplete)

Waiting for a log switch because the session cannot wrap into the next log. Wrapping cannot be performed because the checkpoint for that log has not completed.

Wait Time: 1 second

Parameters: None

C.3.99 log file switch (clearing log file)

Waiting for a log switch because the log is being cleared due to a `CLEAR LOGFILE` command or implicit clear logfile executed by recovery.

Wait Time: 1 second

Parameters: None

C.3.100 log file switch (private strand flush incomplete)

User sessions trying to generate redo, wait on this event when LGWR waits for DBWR to complete flushing redo from IMU buffers into the log buffer; when DBWR is complete LGWR can then finish writing the current log, and then switch log files.

Wait Time: 1 second

Parameters: None

C.3.101 log file switch completion

Waiting for a log switch to complete.

Wait Time: 1 second

Parameters: None

C.3.102 log file sync

When a user session commits, the session's redo information must be flushed to the redo logfile. The user session will post the LGWR to write the log buffer to the redo log file. When the LGWR has finished writing, it will post the user session.

Wait Time: The wait time includes the writing of the log buffer and the post.

Parameter	Description
buffer#	The number of the physical buffer in the redo log buffer that must be synchronized

See Also:

- *Oracle Database Performance Tuning Guide* for more information about this wait event
- *Oracle Database Performance Tuning Guide* for information about potential causes of this wait event

C.3.103 log switch/archive

Used as part of the `ALTER SYSTEM ARCHIVE LOG CHANGE` *scn* statement. The session waits for the current log from all open threads to be archived.

Wait Time: Wait for up to 10 seconds

Parameter	Description
thread#	The thread number of the thread that is currently archiving its current log

C.3.104 optimizer stats update retry

When concurrent sessions try to update optimizer statistics for the same object, all of them except the one that successfully acquired all necessary locks/pins on the library/row cache entries, wait on this wait event and then retry locking after a short period of time. In addition to explicit statistics gathering and maintenance operations using the `DBMS_STATS` package, Oracle database itself may try to update statistics for some objects, either on behalf of the user or for its own maintenance purposes.

Wait Time: 10 ms

Parameters: None

C.3.105 parallel recovery change buffer free

The parallel recovery coordinator is waiting for a change mapping buffer to be released by one of the recovery workers.

Wait Time: 100ms

Parameters: None

C.3.106 parallel recovery control message reply

The parallel recovery coordinator is waiting for all recovery workers to respond to a synchronous control message.

Wait Time: 100ms

Parameters: None

C.3.107 parallel recovery coord send blocked

The parallel recovery coordinator cannot send a redo change message because the recovery worker is still actively applying redo that it has already received and has not yet released the channel.

Wait Time: 100ms

Parameters: None

C.3.108 parallel recovery coord wait for reply

The parallel recovery coordinator is waiting for all recovery workers to exit.

Wait Time: 100ms

Parameters: None

C.3.109 parallel recovery coordinator waits for slave cleanup

The parallel recovery coordinator is waiting for all recovery workers to exit gracefully.

Wait Time: 10ms

Parameters: None

C.3.110 parallel recovery read buffer free

The parallel recovery coordinator is waiting for a log read buffer to be released by all recovery workers. Only after every recovery worker finishes applying redo from a log read buffer, can the buffer be used by the coordinator to issue the next log read.

Wait Time: 100ms

Parameters: None

C.3.111 parallel recovery slave next change

A parallel recovery worker is idle and waiting for the next change message from the coordinator.

Wait Time: 100ms

Parameters: None

C.3.112 pending global transaction(s)

This event should happen only during testing. The session waits for pending transactions to clear.

Wait Time: 30 seconds

Parameter	Description
scans	Number of times the session has scanned the `PENDING_TRANS$` table

C.3.113 pipe get

The session waits for a message to be received on the pipe or for the pipe timer to expire.

Wait Time: There is a 5 second wake up (check) and the pipe timer set by the user

Parameter	Description
handle address	The library cache object handle for this pipe
buffer length	The length of the buffer
timeout	The pipe timer set by the user

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.114 pipe put

The session waits for the pipe send timer to expire or for space to be made available in the pipe.

Wait Time: There is the 5 second wakeup (check) and the user-supplied timeout value

Parameter	Description
handle address	The library cache object handle for this pipe
record length	The length of the record or buffer that has been put into the pipe
timeout	The pipe timer set by the user

C.3.115 PL/SQL lock timer

This event is called through the `DBMS_LOCK.SLEEP` procedure. This event will most likely originate from procedures written by a user.

Wait Time: The wait time is in hundredths of seconds and depends on the user context

Parameter	Description
duration	The duration that the user specified in the `DBMS_LOCK.SLEEP` procedure

C.3.116 pmon timer

This is the main wait event for PMON. When PMON is idle, it is waiting on this event.

Wait Time: Up to 3 seconds, if not posted before

Parameter	Description
duration	The actual amount of time that the PMON is trying to sleep

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.117 prior spawner clean up

When a prior process has terminated while spawning a background, the current process which is trying to spawn new a background must wait until the prior process state is cleaned up.

Wait Time: Usually 3 - 10 seconds

Parameter	Description
process_pid	process identifier (see `V$PROCESS.PID`) of the process whose state must be cleaned up.
process_sno	process serial number (see `V$PROCESS.SERIAL#`) of the process whose state must be cleaned up.

C.3.118 process startup

Wait for a shared server, Dispatcher, or other background process to start.

Wait Time: Wait up to 1 second for a background process to start. If timed out, then re-wait until 5 minutes have passed and signal an error. If the process has started, the event will acknowledge this.

Parameter	Description
type	The process type that was started
process#	The process number of the process being started
waited	Cumulative time waited for the process to start

C.3.119 PX Deque wait

The process is waiting for a message during a parallel execute.

Wait Time: The wait time depends on how quickly the message arrives. Wait times can vary, but it will normally be a short period of time.

Parameter	Description
reason	The reason for dequeuing
sleeptime	The amount of time that the session slept
loop	The total number of times that the session has slept

C.3.120 PX qref latch

Each parallel execution process has a parallel execution qref latch, which must be acquired before the queue buffers can be manipulated.

Wait Time: Wait up to 1 second

Parameter	Description
function	Indicates the type of wait that the session is doing
sleeptime	The amount of time that the session waits (in hundredths of a second)
qref	The address of the process queue for which the session is waiting

C.3.121 PX server shutdown

During normal or immediate shutdown the parallel execution servers are posted to shutdown cleanly. If any parallel execution servers are still running after 10 seconds, they are terminated.

Wait Time: Wait up to 0.5 seconds

Parameter	Description
nalive	The number of parallel execution servers that are still running
sleeptime	The total sleeptime since the session started to wait on this event
loop	The number of times the session waited for this event

C.3.122 PX signal server

This event occurs only in Exclusive mode. The query coordinator is signaling the Query Workers that an error has occurred.

Wait Time: 0.5 seconds

Parameter	Description
serial	The serial number of the worker process queue
error	The error that has occurred
nbusy	The number of worker processes that are still busy

C.3.123 rdbms ipc message

The background processes (LGWR, DBWR, LMS0) use this event to indicate that they are idle and are waiting for the foreground processes to send them an IPC message to do some work.

Wait Time: Up to 3 seconds. The parameter *timeout* shows the true sleep time.

Parameter	Description
timeout	The amount of time that the session waits for an IPC message

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.124 rdbms ipc message block

This event indicates that all message blocks are in use and that the session had to wait for a message block to become available.

Wait Time: Wait up to 60 seconds

Parameters: None

C.3.125 rdbms ipc reply

This event is used to wait for a reply from one of the background processes.

Wait Time: The wait time is specified by the user and is indicated by the parameter *timeout*.

Parameter	Description
from_process	The background process for which the session is waiting. The wait is for a reply to an IPC message sent by the session.
timeout	The amount of time in seconds that this process will wait for a reply

C.3.126 read by other session

This event occurs when a session requests a buffer that is currently being read into the buffer cache by another session. Prior to release 10.1, waits for this event were grouped with the other reasons for waiting for buffers under the 'buffer busy waits' event.

Wait Time: Time waited for the buffer to be read by the other session (in microseconds)

Parameter	Description
file#	See "file#"
block#	See "block#"
class#	See "class"

C.3.127 recovery active instance mapping setup

This event is used to track the time taken (in centiseconds) to set up recovery context on remote instances.

C.3.128 recovery apply pending

This event tracks the time the logmerger process waited (in centiseconds) for apply workers to apply all pending changes up to a certain SCN.

C.3.129 recovery cancel

This wait event is used to track the time taken (in centiseconds) to cancel multi instance media recovery.

C.3.130 recovery checkpoint

This wait event tracks the time (in centiseconds) spent waiting for DBWR processes on all instances to write dirty buffers to disk up to a certain SCN.

C.3.131 recovery file header update for checkpoint

This wait event tracks the time (in centiseconds) spent by MRP0 process waiting on logmerger to update file headers to record the latest checkpoint.

C.3.132 recovery file header update for fuzziness

This wait event tracks the time taken (in centiseconds) to update file headers to record media recovery fuzziness.

C.3.133 recovery marker apply

This wait event is used to track the time (in centiseconds) taken to process a recovery marker.

C.3.134 recovery merge pending

This wait event is used to track the time spent (in centiseconds) by the MRP0 process waiting for remote logmergers to merge changes up to a certain SCN.

C.3.135 recovery metadata latch

This wait event tracks the time taken (in centiseconds) to get the active DataGuard metadata update latch to advance a query SCN.

C.3.136 recovery move influx buffers

This wait event is used to track the time spent (in centiseconds) waiting for all apply workers to move their influx buffers to the dirty queue.

C.3.137 recovery read

A parallel recovery worker (or serial recovery process) is waiting for a batch of data block reads to complete.

Wait Time: Time it takes to complete the physical I/O (read)

Parameters: None

C.3.138 recovery receive buffer free

This wait event is used to track the time (in centiseconds) spent by the receiver process on instance waiting for apply workers to apply changes from received buffers so that they can be freed for the next change.

C.3.139 recovery remote file verification

This wait event tracks the time spent (in centiseconds) by the MRP0 process waiting for remote logmergers to finish file verification on remote instances.

C.3.140 recovery send buffer free

This wait event is used to track the time spent (in centiseconds) by the sender process waiting for apply workers to apply changes from the local instance so that the buffer can be reused for new changes.

C.3.141 recovery shutdown

This wait event is used to track the time (in centiseconds) taken for remote logmergers to shut down.

C.3.142 Redo Transport Attach

Used to track the time spent (in centiseconds) doing Connect, Logon, and RFSATTACH for *any* network process.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.143 Redo Transport Close

Used to track the time spent (in centiseconds) by ARC*n*, NSS*n*, and TT*nn* processes doing RFSCLOSE and RFSRGSTR operations.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.144 Redo Transport Detach

Used to track the time spent (in centiseconds) doing RFSDETACH and Disconnect for *any* network process.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.145 Redo Transport Open

Used to track the time spent (in centiseconds) by ARC*n*, NSS*n*, and TT*nn* background processes doing RFSCREAT and RFSANNCE operations.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.146 Redo Transport Ping

Used to track the time spent (in centiseconds) by ARC*n* background processes doing RFSPING operations.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.147 Redo Transport Slave Startup

Used to track the time spent (in centiseconds) by LGWR doing NSS*n* and TT*nn* process startup and initialization.

C.3.148 Redo Writer Remote Sync Complete

Used to track the time spent (in centiseconds) by LGWR reaping completed network writes to remote destinations.

C.3.149 Redo Writer Remote Sync Notify

Used to track the time spent (in centiseconds) by LGWR issuing network writes to remote destinations.

C.3.150 resmgr:become active

The session is waiting for a resource manager active session slot. This event occurs when the resource manager is enabled and the number of active sessions in the session's current consumer group exceeds the current resource plan's active session limit for the consumer group. To reduce the occurrence of this wait event, increase the active session limit for the session's current consumer group.

Wait Time: The time the session waited to be allocated an active session slot

Parameter	Description
location	location of the wait

C.3.151 resmgr:cpu quantum

The session is waiting to be allocated a quantum of CPU. This event occurs when the resource manager is enabled and is throttling CPU consumption. To reduce the occurrence of this wait event, increase the CPU allocation for the sessions's current consumer group.

Wait Time: The time the session waited to acquire a CPU quantum

Parameter	Description
location	Location of the wait
Consumer Group ID	Consumer group ID of the session waiting. This value is from the `CONSUMER_GROUP_ID` column in the `DBA_RSRC_CONSUMER_GROUPS` view.

> **See Also:**
>
> "DBA_RSRC_CONSUMER_GROUPS"

C.3.152 resmgr: I/O rate limit

The session is waiting for I/O bandwidth to be available to issue I/Os. This event occurs when PDB I/O rate limits are enabled by setting the `MAX_IOPS` and/or `MAX_MBPS` parameters.

To reduce the occurrence of this wait event, the PDB I/O rate limit values need to be increased. This can be achieved by increasing the values of `MAX_IOPS` and `MAX_MBPS` or by removing the limits altogether by setting the parameter value to 0.

Wait Time: The time the session waited for the I/O bandwidth to become available.

Parameter	Description
P1: PDB_ID	This the PDB ID of the PDB that has its I/O throttled because the `MAX_IOPS` or `MAX_MBPS` parameter was set for it.

> **See Also:**
>
> - "MAX_IOPS"
> - "MAX_MBPS"

C.3.153 resmgr:pq queued

The session is waiting in the parallel statement queue.

Wait Time: The time the session waited for sufficient parallel query processes to become available to run this session with the requested degree of parallelism

Parameter	Description
location	Location of the wait

> **See Also:**
>
> *Oracle Database VLDB and Partitioning Guide* for more information about this wait event

C.3.154 rolling migration: cluster quiesce

This is the wait event that instances wait on when cluster is about to start a rolling migration. The instances are waiting for any privileged operations that blocks rolling migration to complete before allowing rolling migration.

Wait Time: 1 second

Parameter	Description
location	Its value will be 1 if the wait is for completion of the privileged operations so that a rolling upgrade/downgrade can start. Its value will be 2 if the wait is for completion of the rolling upgrade/downgrade on all the nodes in the cluster.
waits	The number of seconds spent waiting at the current location.

C.3.155 row cache lock

The session is trying to get a data dictionary lock.

Wait Time: Wait up to 60 seconds.

Parameter	Description
cache id	The `CACHE#` column value in the `V$ROWCACHE` view
mode	See "mode"
request	The pipe timer set by the user

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.156 RVWR wait for flashback copy

Waits for a process to copy flashback database data into the flashback buffer, in order to write out the requested flashback data.

Wait Time: 10 milliseconds

Parameter	Description
copy latch #	The number of flashback copy latch RVWR is waiting on

C.3.157 sbtbufinfo

This function is called when Oracle needs to discover the size, and number, of I/O buffers that have been allocated by the SBT layer. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

C.3.158 sbtgetbuf

This function obtains one I/O buffer that Oracle will use for I/O during a backup job.

Wait Time: Less than one millisecond

Parameters: None

C.3.159 sbtmapbuf

This is an internal function used to facilitate multi-process buffer management. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

C.3.160 sbtrelbuf

This function releases an I/O buffer that has been already processed during a restore job, so that the SBT layer can fill it with more data. It should be very fast and never block.

Wait Time: Less than one millisecond

Parameters: None

C.3.161 scginq AST call

Called by the session to find the highest lock mode that is held on a resource.

Wait Time: Wait up to 0.2 seconds, but the wait will continue until the NULL mode Acquisition AST has fired.

Parameters: None

C.3.162 SGA: allocation forcing component growth

Process waiting on an immediate mode memory transfer with auto-tune SGA after a 4031 for MMAN to get the memory and post it.

Wait Time: 10 msec

Parameters: None

C.3.163 SGA: MMAN sleep for component shrink

MMAN to wait and post itself for satisfying an auto-tuned memory request while trying to fully free a component's quiesced granules. In Release 10.1, the name of this event was 'wait for SGA component shrink'.

Wait Time: 10 msec

Parameter	Description
P1	component_id (corresponding to the memory pool)
P2	Current size in granules
P3	Target size in granules

C.3.164 SGA: sga_target resize

Memory resize requests wait while sga target is being resized. In Release 10.1, the name of this event was 'wait for sga_target resize'.

Wait Time: 10 msec

Parameters: None

C.3.165 Shared IO Pool Memory

Wait until a shared I/O pool buffer becomes available. This happens when processes are using these buffers for I/O and the current process needs to wait for the release of any one of the buffers to the shared I/O pool.

Wait Time: 10msec

Parameters: None

C.3.166 shared server idle wait

Idle wait event for a shared server.

The server waits on the common queue for a virtual circuit. (See also "virtual circuit wait".)

Wait Time: 30 seconds

Parameters: None

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.167 single-task message

When running single task, this event indicates that the session waits for the client side of the executable.

Wait Time: Total elapsed time that this session spent in the user application

Parameters: None

C.3.168 smon timer

This is the main idle event for SMON. SMON will be waiting on this event most of the time until it times out or is posted by another process.

Wait Time: 5 minutes (300 seconds)

Parameter	Description
sleeptime	The amount of time that SMON tries to wait on this event in seconds
failed	The number of times SMON was posted when there some kind of error

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.169 SQL*Net break/reset to client

The server sends a break or reset message to the client. The session running on the server waits for a reply from the client.

Wait Time: The actual time it takes for the break or reset message to return from the client

Parameter	Description
driver id	See "driver id"
break?	See "break?"

C.3.170 SQL*Net break/reset to dblink

Same as **SQL*Net break/reset to client**, but in this case, the break/reset message is sent to another server process over a database link.

Wait Time: The actual time it takes for the break or reset message to return from the other server process

Parameter	Description
driver id	See "driver id"
break?	See "break?"

C.3.171 SQL*Net message from client

The server process (foreground process) waits for a message from the client process to arrive.

Wait Time: The time it took for a message to arrive from the client since the last message was sent to the client

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes received by the server (foreground process) from the client.

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.172 SQL*Net message from dblink

The session waits while the server process (foreground process) receives messages over a database link from another server process.

Wait Time: The time it took for a message to arrive from another server (foreground process) since a message was sent to the other foreground process.

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes received by the server (foreground process) from another foreground process over a database link.

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.173 SQL*Net message to client

The server (foreground process) is sending a message to the client.

Wait Time: The actual time the **send** takes

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes sent by the server process to the client

C.3.174 SQL*Net message to dblink

The server process (foreground process) is sending a message over a database link to another server process.

Wait Time: The actual time the **send** takes

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes sent by the server process to another server process over a database link

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.175 SQL*Net more data from client

The server is waiting on the client to send more data to its client shadow process, in an already initiated operation.

Wait Time: The time waited depends on the time it took to receive the data (including the waiting time)

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes received from the client

C.3.176 SQL*Net more data from dblink

The foreground process is expecting more data from a data base link.

Wait Time: The total time it takes to read the data from the database link (including the waiting time for the data to arrive)

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes received

C.3.177 SQL*Net more data to client

The server process is sending more data/messages to the client. The previous operation to the client was also a **send**.

Wait Time: The actual time it took for the **send** to complete

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes that are being sent to the client

> **See Also:**
>
> *Oracle Database Performance Tuning Guide* for more information about this wait event

C.3.178 SQL*Net more data to dblink

The event indicates that the server is sending data over a database link again. The previous operation over this database link was also a **send**.

Wait Time: The actual time it takes to send the data to the other server

Parameter	Description
driver id	See "driver id"
#bytes	The number of bytes that are sent over the database link to the other server process

C.3.179 Streams AQ: waiting for messages in the queue

The session is waiting on an empty OLTP queue (Advanced Queuing) for a message to arrive so that the session can dequeue that message.

Wait Time: The amount of time that the session wants to wait is determined by the parameter *wait time*

Parameter	Description
queue id	The ID of the OLTP queue for which this session is waiting
process#	The process number of the process in which this session runs
wait time	The intended wait time for this session

C.3.180 switch logfile command

The session waits on the user command `SWITCH LOGFILE` to complete.

Wait Time: 5 seconds

Parameters: None

C.3.181 SYNC Remote Write

Used to track the time spent by LGWR doing SYNC RFSWRITE operations.

> **See Also:**
>
> *Oracle Data Guard Concepts and Administration* for more information about using RFS to manage standby redo logs.

C.3.182 TCP Socket (KGAS)

A session is waiting for an external host to provide requested data over a network socket. The time that this wait event tracks does not indicate a problem, and even a long wait time is not a reason to contact Oracle Support. It naturally takes time for data to flow between hosts over a network, and for the remote aspect of an application to process any request made to it.

An application that communicates with a remote host must wait until the data it will read has arrived. In addition, on Microsoft Windows, a separate thread monitors the arrival of traffic. This thread spends most of its life in waits tracked by the TCP Socket (KGAS) wait event.

Wait Time: The total elapsed time for the network connection to be established or for data to arrive from over the network

Parameter	Description
P0	For Oracle internal use only. Values 8, 9, and 10 occur within the special thread present on Microsoft Windows; other P0 values occur in normal user sessions.
P1	For Oracle internal use only

C.3.183 Text batch free

Waiting for any of the Oracle Text index in-memory postings lists batches to free up. Batches free up when they are written to Oracle Text internal index tables by `IR`*`nn`* background processes.

Wait Time: 9 seconds maximum (3 seconds per wait or until posted by an `IR`*`nn`* process)

Parameters: None

C.3.184 Text sync complete

Waiting for an Oracle Text index to complete a background sync in the `CTX_DDL.SYNC_INDEX()` procedure.

Wait Time: Unlimited (3 seconds per wait or until posted by an `IRnn` process)

Parameter	Description
objn	Object number of the Oracle Text index or index partition
location	ID of the PDB in which the wait is occuring

C.3.185 timer in sksawat

The session waits for the Archiver (ARC*n*) asynchronous I/O to complete.

Wait Time: 0.01 seconds

Parameters: None

C.3.186 transaction

Wait for a blocking transaction to be rolled back. Continue waiting until the transaction has been rolled back.

Wait Time: 1 second

Parameter	Description
undo seg#	The rollback segment ID
slot#	The slot ID inside the rollback segment
wrap#	The sequence number that is incremented for each transaction
count	The number of times that the session has waited on this transaction

C.3.187 unbound tx

The session waits to see if there are any transactions that have been started but do not have a Rollback Segment associated with them.

Wait Time: 1 second

Parameters: None

C.3.188 undo_retention publish retry

This wait can occur for two reasons. A session issuing an `ALTER SYSTEM SET UNDO_RETENTION` may wait on this event wait while a cluster reconfiguration takes place. Or the background process MMNL may wait for cluster reconfiguration while attempting to determine the max `UNDO_RETENTION`.

Wait time: 1 second

Parameter	Description
P1	Identifies where the retry is happening. Id = 1 retry while publishing into the max undo_retention namespace. Id = 2 retry while iterator accessing the max undo_retention namespace.
P2	Retry count (maximum number of retries is 5)

C.3.189 undo segment extension

The undo segment is being extended or shrunk. The session must wait until the operation on the undo segment has finished.

Wait Time: 0.01 seconds

Parameter	Description
segment#	The ID of the rollback segment that is being extended or shrunk

C.3.190 undo segment recovery

PMON is rolling back a terminated transaction. The wait continues until rollback finishes.

Wait Time: 3 seconds

Parameter	Description
segment#	The ID of the rollback segment that contains the transaction that is being rolled back
tx flags	The transaction flags (options) set for the transaction that is being rolled back

C.3.191 undo segment tx slot

Wait for a transaction slot to become available within the selected rollback segment. Continue waiting until the slot is available.

Wait Time: 1 second

Parameter	Description
segment#	The ID of the rollback segment that contains the transaction that is being rolled back

C.3.192 utl_file I/O

Waits associated with operations performed using the `UTL_FILE` package.

C.3.193 virtual circuit wait

The session waits for a virtual circuit operation to complete.

Wait Time: 30 seconds

Parameter	Description
circuit#	Indicates the virtual circuit# being waited on
type	Indicates the type of operation the session is waiting for

C.3.194 WCR: replay client notify

During replay, the Workload Replay Client always keeps an open connection to the database to detect some special errors. This session will normally be in the wait state until some specific replay-related exception occurs.

Wait Time: Waits until notified. Times out every 30 seconds.

Parameter	Description
who am I	Identifies the reason for wait for the admin thread. 1 - waiting for REPLAY.START() to be issued. 2 - waiting for the replay to run to completion or REPLAY.CANCEL() to be issued.

C.3.195 WCR: replay clock

A session will wait on this event during replay if it has some logical dependencies on another session that has not yet committed its work.

Wait Time: Depends upon the amount of row lock contention in the original capture

Parameter	Description
wait for scn's hi 4 bytes	High 4 bytes of the 8-byte replay SCN that the session is waiting on
wait for scn's lo 4 bytes	Low 4 bytes of the 8-byte replay SCN that the session is waiting on

C.3.196 WCR: replay lock order

A session will wait on this event during replay if it saw some lock contention during capture.

Wait Time: Depends upon the amount of row lock contention in the original capture

Parameter	Description
wait for scn's hi 4 bytes	High 4 bytes of the 8-byte replay SCN that the session is waiting on
wait for scn's lo 4 bytes	Low 4 bytes of the 8-byte replay SCN that the session is waiting on

C.3.197 WCR: replay paused

When the user issues a DBMS_WORKLOAD_REPLAY.PAUSE_REPLAY command, all the replayed sessions are waiting on this wait event until replay is resumed.

Parameters: None

C.3.198 WCR: Sync context busy

In the default replay mode, commits during replay are synchronized to follow the same order as capture. All commits are serialized using the Sync context. A replayed session will wait on this event if it is trying to commit at the same time as another replayed session.

Parameters: None

C.3.199 write complete waits

The session waits for a buffer to be written. The write is caused by normal aging or by a cross-instance call.

Wait Time: 1 second

Parameter	Description
file#	The rollback segment id that contains the transaction that is being rolled back
block#	The transaction flags (options) set for the transaction that is being rolled back
id	Identifies the reason for waiting