Oracle® Database Reference
12c Release 1 (12.1)
E17615-24

Statistics Descriptions

This section describes the statistics stored in the `V$SESSTAT` and `V$SYSSTAT` views. The statistics are listed here in alphabetical order.

The `CLASS` column contains a number representing one or more statistics classes. The following class numbers are additive:

- 1, User
- 2, Redo
- 4, Enqueue
- 8, Cache
- 16, OS
- 32, Real Application Clusters
- 64, SQL
- 128, Debug

For example, a class value of 72 represents a statistic that relates to SQL statements and caching.

Some statistics are populated only if the `TIMED_STATISTICS` initialization parameter is set to `true`. Those statistics are flagged with a `Y` in the right-hand column.

Table E-1 Database Statistics Descriptions

Name	Class	Description	TIMED_STATISTICS
application wait time	1	The total wait time (in microseconds) for waits that belong to the Application wait class	
background checkpoints completed	8	Number of checkpoints completed by the background process. This statistic is incremented when the background process successfully advances the thread checkpoint.	
background checkpoints started	8	Number of checkpoints started by the background process. This statistic can be larger than "background checkpoints completed" if a new checkpoint overrides an incomplete checkpoint or if a checkpoint is currently under way. This statistic includes only checkpoints of the redo thread. It does not include: • Individual file checkpoints for operations such	

		as offline or begin backup • Foreground (user-requested) checkpoints (for example, performed by `ALTER SYSTEM CHECKPOINT LOCAL` statements)	
background timeouts	128	This is a count of the times where a background process has set an alarm for itself and the alarm has timed out rather than the background process being posted by another process to do some work.	
branch node splits	128	Number of times an index branch block was split because of the insertion of an additional value	
buffer is not pinned count	72	Number of times a buffer was free when visited. Useful only for internal debugging purposes.	
buffer is pinned count	72	Number of times a buffer was pinned when visited. Useful only for internal debugging purposes.	
bytes received via SQL*Net from client	1	Total number of bytes received from the client over Oracle Net Services	
bytes received via SQL*Net from dblink	1	Total number of bytes received from a database link over Oracle Net Services	
bytes sent via SQL*Net to client	1	Total number of bytes sent to the client from the foreground processes	
bytes sent via SQL*Net to dblink	1	Total number of bytes sent over a database link	
Cached Commit SCN referenced	128	Useful only for internal debugging purposes	
calls to get snapshot scn: kcmgss	32	Number of times a snapshot system change number (SCN) was allocated. The SCN is allocated at the start of a transaction.	
calls to kcmgas	128	Number of calls to routine kcmgas to get a new SCN	
calls to kcmgcs	128	Number of calls to routine kcmgcs to get a current SCN	
calls to kcmgrs	128	Number of calls to routine kcsgrs to get a recent SCN	
change write time	8	Elapsed redo write time for changes made to `CURRENT` blocks in 10s of milliseconds.	Y
cleanouts and rollbacks - consistent read gets	128	Number of consistent gets that require both block rollbacks and block cleanouts. **See Also:** "consistent gets"	
cleanouts only - consistent read gets	128	Number of consistent gets that require only block cleanouts, no rollbacks. **See Also:** "consistent gets"	
cluster key scan block gets	64	Number of blocks obtained in a cluster scan	
cluster key scans	64	Number of cluster scans that were started	
cluster wait time	1	The total wait time (in microseconds) for waits that	

		belong to the Cluster wait class
cold recycle reads	8	Number of buffers that were read through the least recently used end of the recycle cache with fast aging strategy
commit cleanout failures: block lost	8	Number of times Oracle attempted a cleanout at commit but could not find the correct block due to forced write, replacement, or switch `CURRENT`
commit cleanout failures: buffer being written	8	Number of times Oracle attempted a cleanout at commit, but the buffer was currently being written
commit cleanout failures: callback failure	8	Number of times the cleanout callback function returns `FALSE`
commit cleanout failures: cannot pin	8	Total number of times a commit cleanout was performed but failed because the block could not be pinned
commit cleanout failures: hot backup in progress	8	Number of times Oracle attempted block cleanout at commit during hot backup. The image of the block must be logged before the buffer can be made dirty.
commit cleanout failures: write disabled	8	Number of times a cleanout block at commit was performed but the writes to the database had been temporarily disabled
commit cleanouts	8	Total number of times the cleanout block at commit function was performed
commit cleanouts successfully completed	8	Number of times the cleanout block at commit function completed successfully
commit nowait performed	1	The number of asynchronous commits that were actually performed. These commits did not wait for the commit redo to be flushed and be present on disk before returning.
commit nowait requested	1	The number of no-wait commit or asynchronous commit requests that were made either using SQL or the OCI transaction control API
Commit SCN cached	128	Number of times the system change number of a commit operation was cached
commit wait/nowait performed	1	The number of asynchronous/synchronous commits that were actually performed
commit wait/nowait requested	1	The number of no-wait or wait commits that were made either using SQL or the OCI transaction control API
commit wait performed	1	The number of synchronous commits that were actually performed. These commits waited for the commit redo to be flushed and be present on disk before returning.
commit wait requested	1	The number of waiting or synchronous commit requests that were made either using SQL or the OCI

		transaction control API	
concurrency wait time	1	The total wait time (in microseconds) for waits that belong to the Concurrency wait class	
consistent changes	8	Number of times a user process has applied rollback entries to perform a consistent read on the block Work loads that produce a great deal of consistent changes can consume a great deal of resources. The value of this statistic should be small in relation to the "consistent gets" statistic.	
consistent gets	8	Number of times a consistent read was requested for a block. **See Also:** "consistent changes" and "session logical reads" statistics	
consistent gets direct	8	Number of times a consistent read was requested for a block bypassing the buffer cache (for example, direct load operation). This is a subset of "consistent gets" statistics value.	
consistent gets from cache	8	Number of times a consistent read was requested for a block from buffer cache. This is a subset of "consistent gets" statistics value.	
CPU used by this session	1	Amount of CPU time (in 10s of milliseconds) used by a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added. A similar problem can exist in the reporting by the operating system, especially on systems that suffer from many context switches.	Y
CPU used when call started	128	The CPU time used when the call is started **See Also:** "CPU used by this session"	Y
CR blocks created	8	Number of `CURRENT` blocks cloned to create CR (consistent read) blocks. The most common reason for cloning is that the buffer is held in a incompatible mode.	
current blocks converted for CR	8	Number `CURRENT` blocks converted to CR state	
cursor authentications	128	Number of privilege checks conducted during execution of an operation	
data blocks consistent reads - undo records applied	128	Number of undo records applied to data blocks that have been rolled back for consistent read purposes	
db block changes	8	Closely related to "consistent changes", this statistic counts the total number of changes that were part of an update or delete operation that were made to all blocks in the SGA. Such changes generate redo log entries and hence become permanent changes to the database if the transaction is committed.	

		This approximates total database work. This statistic indicates the rate at which buffers are being dirtied (on a per-transaction or per-second basis, for example).
db block gets	8	Number of times a `CURRENT` block was requested **See Also:** "consistent gets"
db block gets direct	8	Number of times a `CURRENT` block was requested bypassing the buffer cache (for example, a direct load operation). This is a subset of "db block gets" statistics value.
db block gets from cache	8	Number of times a `CURRENT` block was requested from the buffer cache. This is a subset of "db block gets" statistics value.
DBWR checkpoint buffers written	8	Number of buffers that were written for checkpoints
DBWR checkpoints	8	Number of times the DBWR was asked to scan the cache and write all blocks marked for a checkpoint or the end of recovery. This statistic is always larger than "background checkpoints completed".
DBWR lru scans	8	Number of times that DBWR scans the LRU queue looking for buffers to write. This count includes scans to fill a batch being written for another purpose (such as a checkpoint).
DBWR revisited being-written buffer	8	Number of times that DBWR tried to save a buffer for writing and found that it was already in the write batch. This statistic measures the amount of "useless" work that DBWR had to do in trying to fill the batch. Many sources contribute to a write batch. If the same buffer from different sources is considered for adding to the write batch, then all but the first attempt will be "useless" because the buffer is already marked as being written.
DBWR transaction table writes	8	Number of rollback segment headers written by DBWR. This statistic indicates how many "hot" buffers were written, causing a user process to wait while the write completed.
DBWR undo block writes	8	Number of rollback segment blocks written by DBWR
DDL statements parallelized	32	Number of DDL statements that were executed in parallel
deferred (CURRENT) block cleanout applications	128	Number of times cleanout records are deferred, piggyback with changes, always current get
DFO trees parallelized	32	Number of times a serial execution plan was converted to a parallel plan
dirty buffers inspected	8	Number of dirty buffers found by the user process while it is looking for a buffer to reuse

DML statements parallelized	32	Number of DML statements that were executed in parallel
enqueue conversions	4	Total number of conversions of the state of table or row lock
enqueue deadlocks	4	Total number of deadlocks between table or row locks in different sessions
enqueue releases	4	Total number of table or row locks released
enqueue requests	4	Total number of table or row locks acquired
enqueue timeouts	4	Total number of table and row locks (acquired and converted) that timed out before they could complete
enqueue waits	4	Total number of waits that occurred during an enqueue convert or get because the enqueue get was deferred
exchange deadlocks	8	Number of times that a process detected a potential deadlock when exchanging two buffers and raised an internal, restartable error. Index scans are the only operations that perform exchanges.
execute count	64	Total number of calls (user and recursive) that executed SQL statements
fbda woken up	128	Number of times the flashback data archive background process was woken up to do archiving
file io wait time	1	Total time spent in wait (in microseconds) for I/O to datafiles, excluding the service time for such I/O. This is cumulative for all I/Os for all datafiles. The service time for one I/O operation is estimated as the minimum time spent in the I/O call seen so far. This service time is subtracted from the time spent in each I/O call to get the wait time for that I/O.
flash cache eviction: aged out	8	Flash cache buffer is aged out of the Database Smart Flash Cache
flash cache eviction: buffer pinned	8	Database Smart Flash Cache buffer is invalidated due to object or range reuse, and so on. The Database Flash Cache Buffer was in use at the time of eviction.
flash cache eviction: invalidated	8	Database Smart Flash Cache buffer is invalidated due to object or range reuse, and so on. The Database Smart Flash Cache buffer was not in use at the time of eviction.
flash cache insert skip: corrupt	8	In-memory buffer was skipped for insertion into the Database Smart Flash Cache because the buffer was corrupted
flash cache insert skip: DBWR overloaded	8	In-memory buffer was skipped for insertion into the Database Smart Flash Cache because DBWR was busy writing other buffers
flash cache insert skip: exists	8	In-memory buffer was skipped for insertion into the

		Database Smart Flash Cache because it was already in the flash cache
flash cache insert skip: modification	8	In-memory buffer was skipped for insertion into the Database Smart Flash Cache because it was being modified
flash cache insert skip: not current	8	In-memory buffer was skipped for insertion into the Database Smart Flash Cache because it was not current
flash cache insert skip: not useful	8	In-memory buffer was skipped for insertion into the Database Smart Flash Cache because the type of buffer was not useful to keep
flash cache inserts	8	Total number of in-memory buffers inserted into the Database Smart Flash Cache
flashback log write bytes	2	Total size in bytes of flashback database data written by RVWR to flashback database logs
flashback log writes	2	Total number of writes by RVWR to flashback database logs
foreground propagated tracked transactions	128	Number of transactions modifying tables enabled for flashback data archive which were archived by a foreground process
free buffer inspected	8	Number of buffers skipped over from the end of an LRU queue to find a reusable buffer. The difference between this statistic and "dirty buffers inspected" is the number of buffers that could not be used because they had a user, a waiter, or were being read or written, or because they were busy or needed to be written after rapid aging out.
free buffer requested	8	Number of times a reusable buffer or a free buffer was requested to create or load a block
gc read wait failures	40	A read wait is when a CR server waits for a disk read to complete before serving a block to another instance. This statistic displays the number of times a read wait ended in failure, that is, after waiting it was unable to serve a block.
gc read wait time	40	A read wait is when a CR server waits for a disk read to complete before serving a block to another instance. This statistic displays the total time spent waiting for disk reads (in centiseconds). The average time waited is "gc read wait time" / "gc read waits."
gc read wait timeouts	40	A read wait is when a CR server waits for a disk read to complete before serving a block to another instance. This statistic displays the number of times a read wait timed out, that is, the disk read did not complete in time, so the wait was aborted.
gc read waits	40	The number of times a CR server waited for a disk

		read, and then successfully served a block
global cache blocks corrupt	40	Real Application Clusters only: Number of blocks that encountered a corruption or checksum failure during interconnect
global cache convert time	40	Real Application Clusters only: Total time elapsed during lock converts
global cache convert timeouts	40	Number of times lock converts in the global cache timed out
global cache converts	40	Number of lock converts in the global cache
global cache cr block log flushes	40	Number of log flushes of the consistent-read block
global cache cr block log flush time	40	Total time spent by the BSP process in log flushes after sending a constructed consistent-read (CR) block. This statistic divided by "global cache cr blocks served" = log flush time per CR block.
global cache cr block receive time	40	Total amount of time foreground processes waited for a CR block to be sent through the interconnect. This statistic divided by "global cache cr blocks received" = time waited per block.
global cache cr block send time	40	Total time spent by the BSP process in sending constructed consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = send time per CR block.
global cache cr block serve time	40	Total amo unt of time the BSP process took to construct consistent-read (CR) blocks. This statistic divided by "global cache cr blocks served" = construction time per CR block.
global cache cr blocks received	40	Total number of blocks received
global cache cr blocks served	40	Total number of blocks constructed by the BSP process
global cache cr requests blocked	40	Number of times foreground attempt to request a cr block and failed
global cache cr timeouts	40	Number of times a foreground process requested a consistent-read (CR) block when the request timed out
global cache defers	40	Number of times a lock was requested and the holder of the lock deferred the release
global cache freelist waits	40	System configured with fewer lock elements than buffers. Number of times foreground has to wait for a lock element.
global cache get time	40	Total time spent waiting. This divided by global cache gets = time waited per request.
global cache gets	40	Number of locks acquired
global cache prepare failures	40	Number of times a failure occurred during preparation

		for interconnect transfer
global enqueue get time	32	Total elapsed time in 10s of milliseconds of all synchronous and asynchronous global enqueue gets and converts
global enqueue gets async	32	Total number of asynchronous global enqueue gets and converts
global enqueue gets sync	32	Total number of synchronous global enqueue gets and converts
global enqueue releases	32	Total number of synchronous global enqueue releases
global enqueue CPU used by this session	32	Amount of CPU time (in 10s of milliseconds) used by synchronous and asynchronous global enqueue activity in a session from the time a user call starts until it ends. If a user call completes within 10 milliseconds, the start and end user-call time are the same for purposes of this statistics, and 0 milliseconds are added.
hot buffers moved to head of LRU	8	When a hot buffer reaches the tail of its replacement list, Oracle moves it back to the head of the list to keep it from being reused. This statistic counts such moves.
in call idle wait time	1	The total wait time (in microseconds) for waits that belong to the Idle wait class. **See Also:** "non-idle wait count" and "non-idle wait time"
immediate (CR) block cleanout applications	128	Number of times cleanout records are applied immediately during consistent-read requests
immediate (CURRENT) block cleanout applications	128	Number of times cleanout records are applied immediately during current gets. Compare this statistic with "deferred (CURRENT) block cleanout applications"
index fast full scans (direct read)	64	Number of fast full scans initiated using direct read
index fast full scans (full)	64	Number of fast full scans initiated for full segments
index fast full scans (rowid ranges)	64	Number of fast full scans initiated with rowid endpoints specified
instance recovery database freeze count	32	Number of times the database is frozen during instance recovery
kcmccs called get current scn	32	Number of times the kernel got the CURRENT SCN when there was a need to casually confirm the SCN
kcmgss read scn without going to DLM	32	Number of times the kernel got a snapshot SCN without going to the distributed lock manager (DLM)
kcmgss waited for batching	32	Number of times a database process is blocked waiting for a snapshot SCN
large tracked transactions	128	For tables tracked by flashback data archive, the number of transactions modifying rows in those tables

		which are large in terms of size or number of changes
leaf node splits	128	Number of times an index leaf node was split because of the insertion of an additional value
lob reads	8	Number of LOB API read operations performed in the session/system. A single LOB API read may correspond to multiple physical/logical disk block reads.
lob writes	8	Number of LOB API write operations performed in the session/system. A single LOB API write may correspond to multiple physical/logical disk block writes.
lob writes unaligned	8	Number of LOB API write operations whose start offset or buffer size is not aligned to the internal chunk size of the LOB. Writes aligned to chunk boundaries are the most efficient write operations. The internal chunk size of a LOB is available through the LOB API (for example, DBMS_LOB.GETCHUNKSIZE()).
logons cumulative	1	Total number of logons since the instance started. Useful only in V$SYSSTAT. It gives an instance overview of all processes that logged on.
logons current	1	Total number of current logons. Useful only in V$SYSSTAT.
messages received	128	Number of messages sent and received between background processes
messages sent	128	Number of messages sent and received between background processes
native hash arithmetic execute	64	Number of hash operations performed using native arithmetic rather than Oracle NUMBERs
native hash arithmetic fail	64	Number of has operations performed using native arithmetic that failed, requiring the hash operation to be performed with Oracle NUMBERs
next scns gotten without going to DLM	32	Number of system change numbers obtained without going to the distributed lock manager or server
no buffer to keep pinned count	72	Number of times a visit to a buffer attempted, but the buffer was not found where expected. Like "buffer is not pinned count" and "buffer is pinned count", this statistic is useful only for internal debugging purposes.
no work - consistent read gets	128	Number consistent gets that require neither block cleanouts nor rollbacks. **See Also:** "consistent gets"
non-idle wait count	1	The total number of waits performed with wait events that were not part of the Idle wait class. **See Also:** "in call idle wait time" and "non-idle wait time"

non-idle wait time	1	The total wait time (in microseconds) for waits that do not belong to the Idle wait class. **See Also:** "in call idle wait time" and "non-idle wait count"
OLAP Paging Manager Cache Hit	64	The number of times a requested page is found in the OLAP page pool. Use this statistic in conjunction with OLAP Paging Manager Cache Miss to determine the OLAP page pool efficiency ratio.
OLAP Paging Manager Cache Miss	64	The number of times a requested page is not found in the OLAP page pool. Use this statistic in conjunction with OLAP Paging Manager Cache Hit to determine the OLAP page pool efficiency ratio.
OLAP Paging Manager New Page	64	The number of newly-created pages in the OLAP page pool that have not yet been written to the workspace LOB
OLAP Paging Manager Cache Write	64	The number of times the OLAP paging manager writes to a page in the OLAP page pool.
OLAP Session Cache Hit	64	The number of times the requested, dynamically-aggregated value of an AW object, was found in the OLAP session cache.
OLAP Session Cache Miss	64	The number of times the requested, dynamically-aggregated value of an AW object, was not found in the OLAP session cache.
OLAP Aggregate Function Calc	64	The number of times the `AGGREGATE` function computes a parent value based on the values of its children.
OLAP Aggregate Function Precompute	64	The number of times the `AGGREGATE` function is to compute a value and finds it precomputed in the cube.
OLAP Aggregate Function Logical NA	64	The number of times an `AGGREGATE` function evaluates to a logical NA value. This could be because the `AGGINDEX` is on and the composite tuple does not exist.
OLAP Paging Manager Pool Size	64	Size, in bytes, of the OLAP page pool allocated to a session and the sum of all OLAP page pools in the system.
OLAP Import Rows Pushed	64	The number of OLAP import rows pushed. This statistic refers to the number of rows encountered from a source cursor and is useful during cube build operations.
OLAP Import Rows Loaded	64	The number of OLAP import rows loaded. This statistic provides the number of rows of the source cursor that are actually loaded into an Analytic Workspace (AW). The difference between the OLAP Import Rows Pushed and OLAP Import Rows Loaded provides the number of rejected rows.

OLAP Row Source Rows Processed	64	The number of rows processed by the OLAP row source
OLAP Engine Calls	64	The total number of OLAP transactions executed within the session. This value provides a general indication of the level of OLAP activity in the session.
OLAP Temp Segments	64	The number of OLAP pages stored in temporary segments for analytic workspaces
OLAP Temp Segments Read	64	The number of times data was read from a temporary segment and not from the OLAP page pool
OLAP Perm LOB Read	64	The number of times data was read from the table where the AW is stored. These are permanent LOB reads.
OLAP Paging Manager Cache Changed Page	64	The number of times the OLAP page pool is changed for any attached AW.
OLAP Fast Limit	64	The number of times an OLAP table function issues a fast limit
OLAP GID Limit	64	The number of times an OLAP table function issues a Cube Grouping ID (CGID) limit. Typically, this type of limit occurs for query rewrite transformations that resolve to a cube organized materialized view.
OLAP INHIER Limit	64	The number of times an OLAP table function issues an in-hierarchy limit. This type of limit can occur when you use cube dimension hierarchy views.
OLAP Full Limit	64	The number of times an OLAP table function issues a full limit
OLAP Custom Member Limit	64	The number of times an OLAP table function issues a custom member limit
OLAP Row Id Limit	64	The number of times an OLAP table function issues a row Id limit.
OLAP Limit Time	64	The total time taken by all the OLAP Limit operations that were performed during the last call to the OLAP table function
OLAP Row Load Time	64	The total time spent loading rows into an AW during cube build and OLAP SQL import operations. Use this statistic along with the OLAP engine elapsed time to measure time spent running OLAP engine routines that involve loading data into AWs from a SQL source. This statistic has the following levels of precision: • Low precision timer This captures the elapsed time of the entire fetch phase of the SQL cursor that is being loaded into AWs. It includes the SQL execution time that occurs during a fetch operation from a source cursor and time taken by the OLAP

engine to populate AWs.

- High precision timer
  This captures the elapsed time, excluding the SQL processing of the cursor being loaded. It records the time spent in the OLAP engine only.
- Default timer precision:
  This is based on the `STATISTIC_LEVEL` parameter. If the low precision is used, then `STATISTICS_LEVEL` is TYPICAL. The high precision timer is used when `STATISTIC_LEVEL` is set to ALL. No timing is captured when `STATISTICS_LEVEL` is BASIC.

OLAP Unique Key Attribute Limit	64	The number of times an OLAP table function issues a unique key attribute limit
opened cursors cumulative	1	In V$SYSSTAT: Total number of cursors opened since the instance started. In V$SESSTAT: Total number of cursors opened since the start of the session.
opened cursors current	1	Total number of current open cursors
opens of replaced files	8	Total number of files that had to be reopened because they were no longer in the process file cache
opens requiring cache replacement	8	Total number of file opens that caused a current file in the process file cache to be closed
OS All other sleep time	16	Time spent sleeping for reasons other than misses in the data segment (see "OS Data page fault sleep time"), kernel page faults (see "OS Kernel page fault sleep time"), or misses in the text segment (see "OS Text page fault sleep time"). An example of such a reason is expiration of quanta.
OS Chars read and written	16	Number of bytes read and written
OS CPU Qt wait time	1	The time a session spends on the CPU run queue (in microseconds), waiting to get the CPU to run
OS Data page fault sleep time	16	Time spent sleeping due to misses in the data segment
OS Input blocks	16	Number of read I/Os
OS Involuntary context switches	16	Number of context switches that were enforced by the operating system
OS Kernel page fault sleep time	16	Time spent sleeping due to operating system kernel page faults
OS Major page faults	16	Number of page faults that resulted in I/O
OS Messages received	16	Number of messages received
OS Messages sent	16	Number of messages sent
OS Minor page faults	16	Number of page faults that did not result in an actual

		I/O
OS Other system trap CPU time	16	Total amount of time to process system traps (as distinct from system calls)
OS Output blocks	16	Number of write I/Os
OS Process heap size	16	Size of area in memory allocated by the process. Typically this represents memory obtained by way of malloc().
OS Process stack size	16	Size of the process stack segment
OS Signals received	16	Number of signals received
OS Swaps	16	Number of swap pages
OS System call CPU time	16	Total amount of time spent executing in system mode
OS System calls	16	Number of system calls
OS Text page fault sleep time	16	Time spent sleeping due to misses in the text segment
OS User level CPU time	16	Total amount of time spent executing in user mode
OS Voluntary context switches	16	Number of voluntary context switches (for example, when a process gives up the CPU by a SLEEP() system call)
Parallel operations downgraded 1 to 25 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers
Parallel operations downgraded 25 to 50 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers
Parallel operations downgraded 50 to 75 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers
Parallel operations downgraded 75 to 99 pct	32	Number of times parallel execution was requested and the degree of parallelism was reduced because of insufficient parallel execution servers
Parallel operations downgraded to serial	32	Number of times parallel execution was requested but execution was serial because of insufficient parallel execution servers
Parallel operations not downgraded	32	Number of times parallel execution was executed at the requested degree of parallelism
parse count (hard)	64	Total number of parse calls (real parses). A hard parse is a very expensive operation in terms of memory use, because it requires Oracle to allocate a workheap and other memory structures and then build a parse tree.
parse count (describe)	64	Total number of parse calls on a describe cursor. This operation is a less expensive than a hard parse and more expensive than a soft parse.
parse count (total)	64	Total number of parse calls (hard, soft, and describe).

		A soft parse is a check on an object already in the shared pool, to verify that the permissions on the underlying object have not changed.	
parse time cpu	64	Total CPU time used for parsing (hard and soft) in 10s of milliseconds	Y
parse time elapsed	64	Total elapsed time for parsing, in 10s of milliseconds. Subtract "parse time cpu" from the this statistic to determine the total waiting time for parse resources.	Y
physical read bytes	8	Total size in bytes of all disk reads by application activity (and not other instance activity) only.	
physical read flash cache hits	8	Total number of reads from flash cache instead of disk	
physical read IO requests	8	Number of read requests for application activity (mainly buffer cache and direct load operation) which read one or more database blocks per request. This is a subset of "physical read total IO requests" statistic.	
physical read requests optimized	8	Number of read requests that read one or more database blocks from the Database Smart Flash Cache or the Exadata Smart Flash Cache.	
physical read total bytes	8	Total size in bytes of disk reads by all database instance activity including application reads, backup and recovery, and other utilities. The difference between this value and "physical read bytes" gives the total read size in bytes by non-application workload.	
physical read total IO requests	8	Number of read requests which read one or more database blocks for all instance activity including application, backup and recovery, and other utilities. The difference between this value and "physical read total multi block requests" gives the total number of single block read requests.	
physical read total multi block requests	8	Total number of Oracle instance read requests which read in two or more database blocks per request for all instance activity including application, backup and recovery, and other utilities.	
physical reads	8	Total number of data blocks read from disk. This value can be greater than the value of "physical reads direct" plus "physical reads cache" as reads into process private buffers also included in this statistic.	
physical reads cache	8	Total number of data blocks read from disk into the buffer cache. This is a subset of "physical reads" statistic.	
physical reads cache prefetch	8	Number of contiguous and noncontiguous blocks that were prefetched.	
physical reads direct	8	Number of reads directly from disk, bypassing the buffer cache. For example, in high bandwidth, data-intensive operations such as parallel query, reads of disk blocks bypass the buffer cache to maximize	

		transfer rates and to prevent the premature aging of shared data blocks resident in the buffer cache.
physical reads direct (lob)	8	Number of buffers that were read directly for LOBs
physical reads direct temporary tablespace	8	Number of buffers that were read directly from temporary tablespaces
physical reads for flashback new	8	Number of blocks read for newing (that is, preparing a data block for a completely new change) blocks while flashback database is enabled
physical reads prefetch warmup	8	Number of data blocks that were read from the disk during the automatic prewarming of the buffer cache.
physical write bytes	8	Total size in bytes of all disk writes from the database application activity (and not other kinds of instance activity).
physical write IO requests	8	Number of write requests for application activity (mainly buffer cache and direct load operation) which wrote one or more database blocks per request.
physical write total bytes	8	Total size in bytes of all disk writes for the database instance including application activity, backup and recovery, and other utilities. The difference between this value and "physical write bytes" gives the total write size in bytes by non-application workload.
physical write total IO requests	8	Number of write requests which wrote one or more database blocks from all instance activity including application activity, backup and recovery, and other utilities. The difference between this stat and "physical write total multi block requests" gives the number of single block write requests.
physical write total multi block requests	8	Total number of Oracle instance write requests which wrote two or more blocks per request to the disk for all instance activity including application activity, recovery and backup, and other utilities.
physical writes	8	Total number of data blocks written to disk. This statistics value equals the sum of "physical writes direct" and "physical writes from cache" values.
physical writes direct	8	Number of writes directly to disk, bypassing the buffer cache (as in a direct load operation)
physical writes direct (lob)	8	Number of buffers that were directly written for LOBs
physical writes direct temporary tablespace	8	Number of buffers that were directly written for temporary tablespaces
physical writes from cache	8	Total number of data blocks written to disk from the buffer cache. This is a subset of "physical writes" statistic.
physical writes non checkpoint	8	Number of times a buffer is written for reasons other than advancement of the checkpoint. Used as a metric for determining the I/O overhead imposed by setting

		the FAST_START_IO_TARGET parameter to limit recovery I/Os. (Note that FAST_START_IO_TARGET is a deprecated parameter.) Essentially this statistic measures the number of writes that would have occurred had there been no checkpointing. Subtracting this value from "physical writes" gives the extra I/O for checkpointing.	
pinned buffers inspected	8	Number of times a user process, when scanning the tail of the replacement list looking for a buffer to reuse, encountered a cold buffer that was pinned or had a waiter that was about to pin it. This occurrence is uncommon, because a cold buffer should not be pinned very often.	
prefetched blocks aged out before use	8	Number of contiguous and noncontiguous blocks that were prefetched but aged out before use	
process last non-idle time	128	The last time this process executed	Y
PX local messages recv'd	32	Number of local messages received for parallel execution within the instance local to the current session	
PX local messages sent	32	Number of local messages sent for parallel execution within the instance local to the current session	
PX remote messages recv'd	32	Number of remote messages received for parallel execution within the instance local to the current session	
PX remote messages sent	32	Number of remote messages sent for parallel execution within the instance local to the current session	
queries parallelized	32	Number of SELECT statements executed in parallel	
recovery array read time	8	Elapsed time of I/O during recovery	
recovery array reads	8	Number of reads performed during recovery	
recovery blocks read	8	Number of blocks read during recovery	
recovery blocks read for lost write detection	8	Number of blocks read for lost write checks during recovery.	
recovery blocks skipped lost write checks	8	Number of Block Read Records that skipped the lost write check during recovery.	
recursive calls	1	Number of recursive calls generated at both the user and system level. Oracle maintains tables used for internal processing. When Oracle needs to make a change to these tables, it internally generates an internal SQL statement, which in turn generates a recursive call.	
recursive cpu usage	1	Total CPU time used by non-user calls (recursive calls). Subtract this value from "CPU used by this session" to determine how much CPU time was used	

		by the user calls.	
redo blocks checksummed by FG (exclusive)	2	Number of exclusive redo blocks that were checksummed by the generating foreground processes. An exclusive redo block is the one whose entire redo content belongs to a single redo entry.	
redo blocks checksummed by LGWR	2	Number of redo blocks that were checksummed by the LGWR.	
redo blocks written	2	Total number of redo blocks written. This statistic divided by "redo writes" equals number of blocks per write.	
redo buffer allocation retries	2	Total number of retries necessary to allocate space in the redo buffer. Retries are needed either because the redo writer has fallen behind or because an event such as a log switch is occurring.	
redo entries	2	Number of times a redo entry is copied into the redo log buffer	
redo entries for lost write detection	2	Number of times a Block Read Record is copied into the log buffer.	
redo log space requests	2	Number of times the active log file is full and Oracle must wait for disk space to be allocated for the redo log entries. Such space is created by performing a log switch. Log files that are small in relation to the size of the SGA or the commit rate of the work load can cause problems. When the log switch occurs, Oracle must ensure that all committed dirty buffers are written to disk before switching to a new log file. If you have a large SGA full of dirty buffers and small redo log files, a log switch must wait for DBWR to write dirty buffers to disk before continuing. Also examine the **log file space** and **log file space switch** wait events in `V$SESSION_WAIT`	
redo log space wait time	2	Total time waited in centiseconds for available space in the redo log buffer. See also"redo log space requests"	Y
redo ordering marks	2	Number of times that a system change number was allocated to force a redo record to have a higher SCN than a record generated in another thread using the same block	
redo size	2	Total amount of redo generated in bytes	
redo size for lost write detection	2	Total amount of Block Read Records generated in bytes.	
redo synch time	8	Elapsed time of all "redo synch writes" calls in 10s of milliseconds	Y
redo synch writes	8	Number of times the redo is forced to disk, usually for	

		a transaction commit. The log buffer is a circular buffer that LGWR periodically flushes. Usually, redo that is generated and copied into the log buffer need not be flushed out to disk immediately.	
redo wastage	2	Number of bytes wasted because redo blocks needed to be written before they are completely full. Early writing may be needed to commit transactions, to be able to write a database buffer, or to switch logs.	
redo write broadcast ack count	2	Number of times a commit broadcast acknowledgment has not been received by the time when the corresponding log write is completed. This is only for Oracle RAC.	
redo write broadcast ack time	2	Total amount of the latency associated with broadcast on commit beyond the latency of the log write (in microseconds). This is only for Oracle RAC.	Y
redo write time	2	Total elapsed time of the write from the redo log buffer to the current redo log file in 10s of milliseconds	Y
redo writes	2	Total number of writes by LGWR to the redo log files. "redo blocks written" divided by this statistic equals the number of blocks per write	
remote instance undo block writes	40	Number of times this instance wrote a rollback segment so that another instance could read it	
remote instance undo header writes	40	Number of times this instance wrote a undo header block so that another instance could read it	
rollback changes - undo records applied	128	Number of undo records applied to user-requested rollback changes (not consistent-read rollbacks)	
rollbacks only - consistent read gets	128	Number of consistent gets that require only block rollbacks, no block cleanouts. **See Also:** "consistent gets"	
rows fetched via callback	64	Rows fetched via callback. Useful primarily for internal debugging purposes.	
scheduler wait time	1	The total wait time (in microseconds) for waits that belong to the Scheduler wait class	
serializable aborts	1	Number of times a SQL statement in a serializable isolation level had to abort	
session connect time	1	The connect time for the session in 10s of milliseconds. This value is useful only in `V$SESSTAT`. It is the wall clock time since the logon to this session occurred.	Y
session cursor cache count	64	Total number of cursors cached. This statistic is incremented only if `SESSION_CACHED_CURSORS` > 0. This statistic is the most useful in `V$SESSTAT`. If the value for this statistic in `V$SESSTAT` is close to the setting of the `SESSION_CACHED_CURSORS`	

		parameter, the value of the parameter should be increased.
session cursor cache hits	64	Number of hits in the session cursor cache. A hit means that the SQL (including recursive SQL) or PL/SQL statement did not have to be reparsed. Subtract this statistic from "parse count (total)" to determine the real number of parses that occurred.
session logical reads	1	The sum of "db block gets" plus "consistent gets". This includes logical reads of database blocks from either the buffer cache or process private memory.
session pga memory	1	Current PGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.
session pga memory max	1	Peak PGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.
session stored procedure space	1	Amount of memory this session is using for stored procedures
session uga memory	1	Current UGA size for the session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.
session uga memory max	1	Peak UGA size for a session. Useful only in `V$SESSTAT`; it has no meaning in `V$SYSSTAT`.
shared hash latch upgrades - no wait	8	A shared hash latch upgrade is when a hash latch is upgraded from shared mode to exclusive mode. This statistic displays the number of times the upgrade completed immediately.
shared hash latch upgrades - wait	8	A shared hash latch upgrade is when a hash latch is upgraded from shared mode to exclusive mode. This statistics displays the number of times the upgrade did not complete immediately.
shared io pool buffer get failure	128	Number of unsuccessful buffer gets from the shared I/O pool from instance startup time.
shared io pool buffer get success	128	Number of successful buffer gets from the shared I/O pool from instance startup time.
slave propagated tracked transactions	128	Number of transactions modifying tables enabled for flashback data archive which were archived by a slave process
sorts (disk)	64	Number of sort operations that required at least one disk write Sorts that require I/O to disk are quite resource intensive. Try increasing the size of the initialization parameter `SORT_AREA_SIZE`. For more information, see "SORT_AREA_SIZE".
sorts (memory)	64	Number of sort operations that were performed completely in memory and did not require any disk writes You cannot do much better than memory sorts, except

		maybe no sorts at all. Sorting is usually caused by selection criteria specifications within table join SQL operations.
sorts (rows)	64	Total number of rows sorted
SQL*Net roundtrips to/from client	1	Total number of Oracle Net Services messages sent to and received from the client
SQL*Net roundtrips to/from dblink	1	Total number of Oracle Net Services messages sent over and received from a database link
summed dirty queue length	8	The sum of the dirty LRU queue length after every write request. Divide by **write requests** to get the average queue length after write completion.
switch current to new buffer	8	Number of times the CURRENT block moved to a different buffer, leaving a CR block in the original buffer
table fetch by rowid	64	Number of rows that are fetched using a ROWID (usually recovered from an index) This occurrence of table scans usually indicates either non-optimal queries or tables without indexes. Therefore, this statistic should increase as you optimize queries and provide indexes in the application.
table fetch continued row	64	Number of times a chained or migrated row is encountered during a fetch Retrieving rows that span more than one block increases the logical I/O by a factor that corresponds to the number of blocks than need to be accessed. Exporting and re-importing may eliminate this problem. Evaluate the settings for the storage parameters PCTFREE and PCTUSED. This problem cannot be fixed if rows are larger than database blocks (for example, if the `LONG` data type is used and the rows are extremely large).
table scan blocks gotten	64	During scanning operations, each row is retrieved sequentially by Oracle. This statistic counts the number of blocks encountered during the scan. This statistic tells you the number of database blocks that you had to get from the buffer cache for the purpose of scanning. Compare this value with the value of "consistent gets" to determine how much of the consistent read activity can be attributed to scanning.
table scan rows gotten	64	Number of rows that are processed during scanning operations
table scans (cache partitions)	64	Number of range scans performed on tables that have the CACHE option enabled

table scans (direct read)	64	Number of table scans performed with direct read (bypassing the buffer cache)
table scans (long tables)	64	Long (or conversely short) tables can be defined as tables that do not meet the short table criteria as described in table scans (short tables)
table scans (rowid ranges)	64	During parallel query, the number of table scans conducted with specified ROWID ranges
table scans (short tables)	64	Long (or conversely short) tables can be defined by optimizer hints coming down into the row source access layer of Oracle. The table must have the CACHE option set.
total file opens	8	Total number of file opens performed by the instance. Each process needs several files (control file, log file, database file) in order to work against the database.
tracked rows	128	Number of rows modified in tables enabled for flashback data archive
tracked transactions	128	Number of transactions which modified a table enabled for flashback data archive
transaction lock background get time	128	Useful only for internal debugging purposes
transaction lock background gets	128	Useful only for internal debugging purposes
transaction lock foreground requests	128	Useful only for internal debugging purposes
transaction lock foreground wait time	128	Useful only for internal debugging purposes
transaction rollbacks	128	Number of transactions being successfully rolled back
transaction tables consistent read rollbacks	128	Number of times rollback segment headers are rolled back to create consistent read blocks
transaction tables consistent reads - undo records applied	128	Number of undo records applied to transaction tables that have been rolled back for consistent read purposes
Unnecessary process cleanup for SCN batching	32	Total number of times that the process cleanup was performed unnecessarily because the session or process did not get the next batched SCN. The next batched SCN went to another session instead.
user calls	1	Number of user calls such as login, parse, fetch, or execute When determining activity, the ratio of user calls to RPI calls, give you an indication of how much internal work gets generated because of the type of requests the user is sending to Oracle.
user commits	1	Number of user commits. When a user commits a transaction, the redo generated that reflects the

		changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.
user I/O wait time	1	The total wait time (in microseconds) for waits that belong to the User I/O wait class
user rollbacks	1	Number of times users manually issue the `ROLLBACK` statement or an error occurs during a user's transactions
very large tracked transactions	128	For tables tracked by flashback data archive, number of transactions modifying those tables which are very large in terms of size or number of changes
write clones created in background	8	Number of times a background or foreground process clones a `CURRENT` buffer that is being written. The clone becomes the new, accessible `CURRENT` buffer, leaving the original buffer (now the clone) to complete writing.
write clones created in foreground	8	Number of times a background or foreground process clones a `CURRENT` buffer that is being written. The clone becomes the new, accessible `CURRENT` buffer, leaving the original buffer (now the clone) to complete writing.

Need an example? Tell us more.

Reader Comment

Subject

From steve.furlong@oracle.com Anonymous

Comments, corrections, and suggestions are forwarded to authors every week. By submitting, you confirm you agree to the terms and conditions. Use the OTN forums for product questions. For support or consulting, file a service request through My Oracle Support.

Submit

Copyright © 2002, 2014, Oracle and/or its affiliates. All rights reserved.
Legal Notices