Oracle AWR wait events section

Oracle Tips by Burleson Consulting

Wait Events

This section shows a breakdown of the main wait events in the database, and it also shows the wait event details for foreground user processes.

The Wait Events report section displays all wait events that occurred during the snapshot interval. This section contains wait events statistics for only foreground end-user processes. All the IDLE events are placed at the end of this report. The following is a sample of the wait events section.

```
Wait Events  DB/Inst: LSQ/lsq  Snaps: 1355-1356
-> s  - second
-> cs - centisecond -     100th of a second
-> ms - millisecond -    1000th of a second
-> us - microsecond - 1000000th of a second
-> ordered by wait time desc, waits desc (idle events last)

                                                          Avg
                                        Total Wait     wait  Waits
Event                            Wts   Timeouts Time (s)  (ms)  /txn
-------------------------- ------ -------- ------- ------ -----
db file scattered read      23,611        0   3,453   146  36.1
read by other session       44,218        8   3,440    78  67.6
db file sequential read      5,227        0     466    89   8.0
db file parallel write       1,321        0     240   182   2.0
control file parallel write  1,113        0     121   109   1.7
log file parallel write        726        0      47    64   1.1
control file sequential rea    319        0      28    89   0.5
class slave wait                 5        5      26  5116   0.0
log file sync                  368        0      22    59   0.6
```

Oracle has more than 200 specific wait events, and the description for every wait event is available in the Oracle documentation.

Remember the old saying “time takes time” and recognize that any database will inevitably wait for some resources when running. The intention of this report is to provide important information for what a particular database is waiting, but it is only one clue, and you must experience other sections of the AWR report to find the real bottleneck in the system.

The following are the most common causes of wait events:

DB File Scattered Read: This wait event is usually caused by large full table scans. This is normal for DSS systems but is critical for OLTP systems. The DBA should consider the caching of small tables to eliminate file reads. Also, in the OLTP environment, consideration should be given to tuning SQL statements as well.

DB File Sequential Read: A high number of waits for this event indicates possible problems with join operations of SQLs or invocation of non-selective indexes. This wait is caused by a large number of single block reads.

Buffer Busy: This wait is caused by concurrent access to buffers in the buffer cache. This statistic should be correlated with the Buffer Waits section of the AWR report.

Free Buffer: This wait event indicates that Oracle waited many times for a free buffer in the buffer cache. This could be caused by the small size of buffer cache, or a large number of reads which populated the buffer cache with unnecessary data. In this case, the SQLs and buffer contents should be examined. Also, slow work by the Database Writer (DBWR) process could cause such wait event.

Log Buffer Space: This wait event shows that the Log Writer (LGWR) process is not fast enough to free log cache for new blocks. This could be caused by slow log switches, slow disks serving redo logs, or a small size of the redo log buffer.

Latch Free: This wait event is often caused by not using bind variables in SQL statements. This fact is indicated by the library cache latch in the Latches section of the AWR report. There are other latches that can cause this wait event to be high: redo allocation latch , cache buffers LRU chain, cache buffers chain, etc.

The wait events for background processes are separated in the Background Wait Events section. In most cases, Oracle background processes place very little overhead on the system. However, it makes sense to monitor the database’s wait events activity in order to see how they operate.

SEE CODE DEPOT FOR FULL SCRIPTS