|  |  |  |
| --- | --- | --- |
| **How to Use AWR Reports to Diagnose Database Performance Issues (Doc ID 1359094.1)** | [IMG_256](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4%26_afrLoop=269774337128792%20/o%20To%20Bottom)  [To Bottom](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\o To Bottom) | IMG_257 |

IMG_258

IMG_259

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **In this Document**   |  |  | | --- | --- | |  | [Goal](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l GOAL) |  |  |  | | --- | --- | |  | [Best Practices](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section11) |  |  |  | | --- | --- | |  | [Pro-Active Problem Avoidance and Diagnostic Collection](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section12) |  |  |  | | --- | --- | |  | [Ask Questions, Get Help, And Share Your Experiences With This Article](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section13) |  |  |  | | --- | --- | |  | [Solution](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l FIX) |  |  |  | | --- | --- | |  | [AWR Webcast](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section21) |  |  |  | | --- | --- | |  | [Interpretation](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section22) |  |  |  | | --- | --- | |  | [Top 5 Timed Events](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section23) |  |  |  | | --- | --- | |  | [SQL Statistics](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section24) |  |  |  | | --- | --- | |  | [Analysis:](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section25) |  |  |  | | --- | --- | |  | [Other SQL Statistic Sections](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section26) |  |  |  | | --- | --- | |  | [Load Profile](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section27) |  |  |  | | --- | --- | |  | [Instance Efficiency](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section28) |  |  |  | | --- | --- | |  | [Latch Activity](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section29) |  |  |  | | --- | --- | |  | [Notable timed and wait events:](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section210) |  |  |  | | --- | --- | |  | [CPU time events](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section211) |  |  |  | | --- | --- | |  | [Analysis:](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section212) |  |  |  | | --- | --- | |  | [Actions:](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section213) |  |  |  | | --- | --- | |  | [Other Potential CPU related Issues:](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section214) |  |  |  | | --- | --- | |  | [Check to see if other waits follow the high CPU timed event.](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section215) |  |  |  | | --- | --- | |  | [High External CPU usage](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section216) |  |  |  | | --- | --- | |  | [Troubleshooting CPU usage](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section217) |  |  |  | | --- | --- | |  | ['Log file sync' waits](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section218) |  |  |  | | --- | --- | |  | [Buffer busy waits](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section219) |  |  |  | | --- | --- | |  | [Waits for 'Cursor: mutex/pin'](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section220) |  |  |  | | --- | --- | |  | [Troubleshooting Other Issues](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section221) |  |  |  | | --- | --- | |  | [Use of ADDM Reports alongside AWR](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section222) |  |  |  | | --- | --- | |  | [Other AWR reference Articles](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section223) |  |  |  | | --- | --- | |  | [Statspack](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section224) |  |  |  | | --- | --- | |  | [Discuss Diagnosis of Performance Issues Using AWR Reports](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l aref_section225) |  |  |  | | --- | --- | |  | [References](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l REF) |   IMG_260  **APPLIES TO:**  Oracle Financials Accounting Hub - Version 12.1.3 and later  Oracle Database - Enterprise Edition - Version 10.2.0.1 and later  Information in this document applies to any platform.  **GOAL**  IMG_261  IMG_262  This article aims to provide guidance on how to interpret AWR information specifically for Database Performance issues.  Please remember in order to produce reports, access AWR views, or use the diagnostic information from any part of the Automatic Workload Repository, the Diagnostic Pack License is required. This includes the production of AWR reports, ADDM reports and ASH reports even if these are requested by product support or other agencies.  **NOTE:** Oracle Diagnostics Pack (and Oracle Tuning Pack) is available with Enterprise Edition ONLY. For further details of pack licensing see:  Oracle® Database Licensing Information  12c Release 1 (12.1)  Part number E17614-08  Chapter 1 1 Oracle Database Editions  Feature Availability by Edition [http://docs.oracle.com/database/121/DBLIC/options.htm#DBLIC139](http://docs.oracle.com/database/121/DBLIC/options.htm /l DBLIC139 /o Feature Availability by Edition)    **Best Practices**  ***Pro-Active Problem Avoidance and Diagnostic Collection***  Although some problems may be unforeseen, in many cases problems may be avoidable if signs are detected early enough. Additionally, if an issue does occur, it is no use collecting information about that issue after the event. AWR reports are one of the support recommend methods for collecting such diagnostics. For information on suggested uses, other proactive preparations and diagnostics, see:  Document 1482811.1 Best Practices: Proactively Avoiding Database and Query Performance Issues [Document 1477599.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1477599.1) Best Practices Around Data Collection For Performance Issues  **Ask Questions, Get Help, And Share Your Experiences With This Article**  **Would you like to explore this topic further with other Oracle Customers, Oracle Employees, and Industry Experts?**   [Click here](https://community.oracle.com/message/12160644 /o Discussion Thread: Diagnosing Database Perf using AWR Reports [Document ID 1359094.1] /t _blank) to join the discussion where you can ask questions, get help from others, and share your experiences with this specific article.  Discover discussions about other articles and helpful subjects by clicking [here](https://community.oracle.com/community/support/oracle_database/database_tuning /o My Oracle Support Community - Database Tuning /t _blank) to access the main *My Oracle Support Community* page for Database Tuning.  **SOLUTION**  AWR reports are an extremely useful diagnostic tool for the determination of the potential cause of database wide performance issues.   Typically when a performance issue is detected, you would collect an AWR report covering the period of the poor performance. It is best to use a reporting period no longer than 1 hour as otherwise specifics can be lost.   For information regarding collecting AWR reports refer to:  [Document 1363422.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1363422.1) Automatic Workload Repository (AWR) Reports - Start Point  It is also prudent to Gather AWR reports during times when performance is acceptable to provide baselines for comparison when there is a problem.  Ensure that the baseline snapshot duration is the same as the problem duration to facilitate like with like comparison  **NOTE:** We would also recommend using a matched ADDM report initially to give a pointer to the main issues. Reading the corresponding ADDM report as a first step to tuning can save a lot of time because it immediately points at the main user as compared to trying to understand what an AWR report is presenting.  See: [Use of ADDM Reports along side AWR](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l ADDM)  **AWR Webcast**  There are some extremely useful AWR overview webcasts available for download in the Webcast Archive. Particularly:   * "Troubleshooting DB Performance issues with AWR"   You can download this from the following document:  [Document 1456176.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1456176.1) Oracle Database Advisor Webcast Archives  **Interpretation**  When looking to interpret the reasons for poor database performance, we recommend that you work through the checklist in the following article prior to working through this section:  [Document 1628089.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1628089.1) AWR Report Interpretation Checklist for Diagnosing Database Performance Issues  This document provides guidance on some background information to bear in mind when examining the detail of this section that can help by framing the context of the problem.   Since we are looking at a performance issue, our primary concern is what the database is waiting for.  When processes wait, they are being prevented from doing an activity because of some other factor. High waits provide the highest benefit when wait times are reduced and as such are a good focus.  The Top Wait information provides such information and allows us to focus on the main problem areas without wasting time investigating areas that are not causing significant delay.   * **Top 5 Timed Events**   As mentioned, the Top waits section is the most important single section in the whole report being as it quantifies and allows comparison of the primary diagnostic: what each session is waiting for. An example output is provided below:  Top 5 Timed Events                                         Avg %Total  ~~~~~~~~~~~~~~~~~~                                        wait   Call  Event                                 Waits    Time (s)   (ms)   Time Wait Class  ------------------------------ ------------ ----------- ------ ------ ----------  db file scattered read           10,152,564      81,327      8   29.6   User I/O  db file sequential read          10,327,231      75,878      7   27.6   User I/O  CPU time                                         56,207          20.5  read by other session             4,397,330      33,455      8   12.2   User I/O  PX Deq Credit: send blkd             31,398      26,576    846    9.7      Other           -------------------------------------------------------------  The Top 5 Waits section reports on a number of useful topics related to Events. It records the number of waits encountered in the period and the total time spent waiting together with the average time waited for each event. The section is ordered by the %age of the total call time that each Event is responsible for.   Dependent on what is seen in this section, other report sections may need to be referenced in order to quantify or check the findings. For example, the wait count for a particular event needs to be assessed based upon the duration of the reporting period and also the number of users on the database at the time; 10 Million waits in 10 minutes is far more significant than 10 Million in 10 hours, or if shared among 10 users as opposed to 10,000.   In this example report, almost 60% of the time is spent waiting for I/O related reads.     * Event 'db file scattered read ' is typically used when fetching blocks for a full table scan index fast full scan and performs multi-block IO. * Event 'db file sequential read'  is a single block read and is typically engaged for any activity where  multi-block IO is unavailable (for example index reads).   Another 20% of the time is spent waiting for or using CPU time. High CPU usage is often a symptom of poorly tuned SQL (or at least SQL which has potential to take less resource) of which excessive I/O can also be a symptom. More on CPU usage follows later.   Based on this, we would investigate whether these waits indicate a problem or not. If so, resolve the problem, if not, move on to the next wait to determine if that is a potential cause.   There are 2 main reasons why I/O related waits are going to be top of the waits:     * The database is doing lots of reads * The individual reads are slow   The Top 5 events show us information that helps us here :     * Is the database doing lots of reads?:  The section shows > 10 Million reads for each of these events in the period.  Whether this is a lot depends on whether the report duration is 1 hour or 1 minute.  Check the report duration to asses this.  If the reads do seem excessive, then why would the database do a lot of reads?  The database only reads data because the execution of SQL statements has instructed it to do so. To investigate further refer to the [SQL Statistics](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l SQLStats) Section. * Are the individual reads slow?  The section shows waits of <=8 ms for the 2 I/O related events.  Whether this is fast or slow is dependent on the hardware underlying the I/O subsystem, but typically anything under 20 ms is acceptable.   If the I/O was slow, then you can get further information from the 'Tablespace IO Stats ' section: * Tablespace IO Stats                       DB/Inst: VMWREP/VMWREP  Snaps: 1-15 * -> ordered by IOs (Reads + Writes) desc * Tablespace * ------------------------------ * Av      Av     Av                       Av     Buffer Av Buf * Reads Reads/s Rd(ms) Blks/Rd       Writes Writes/s      Waits Wt(ms) * -------------- ------- ------ ------- ------------ -------- ---------- ------ * TS\_TX\_DATA * 14,246,367     283    7.6     4.6  145,263,880    2,883  3,844,161    8.3 * USER * 204,834       4   10.7     1.0   17,849,021      354     15,249    9.8 * UNDOTS1 * 19,725       0    3.0     1.0   10,064,086      200      1,964    4.9 * AE\_TS * 4,287,567      85    5.4     6.7          932        0    465,793    3.7 * TEMP * 2,022,883      40    0.0     5.8      878,049       17          0    0.0 * UNDOTS3 * 1,310,493      26    4.6     1.0      941,675       19         43    0.0 * TS\_TX\_IDX * 1,884,478      37    7.3     1.0       23,695        0     73,703    8.3 * SYSAUX * 346,094       7    5.6     3.9      112,744        2          0    0.0 * SYSTEM        101,771       2    7.9     3.5       25,098        0        653    2.7  Specifically, look for the timing under Rd(ms).  If it is higher than 20 milliseconds per read and reads are high, then you may want to start investigating a potential I/O bottleneck from the os.  **NOTE:** You should ignore relatively idle tablespaces/files as you can get high values due to disk spin-up etc. which are not relevant. If you have an issue with 10 million reads being slow it is unlikely that a tablespace/file with 10 reads has caused the problem!  For further investigation, the following Document may be helpful:  [Document 223117.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=223117.1) Troubleshooting I/O-related waits  Although high waits for 'db file scattered read' and 'db file sequential read' can be I/O related, it is actually more common to find that these waits are relatively 'normal' based on the SQL that the database is being asked to run. In fact, on a well tuned database, you would want these events to be top of the waits, since that would mean that no 'problem' events were there instead!   The trick is being able to assess whether the high waits is indicative of some SQL statements are not using optimal paths (as mentioned earlier) or otherwise.  If there are high waits for 'db file scattered read', then SQL may not be using optimal access paths and so are tending to do Full Table Scans as opposed to  indexes (or there may be missing indexes or not optimal indexes).  Furthermore, high waits for 'db file sequential read' may indicate SQL statements are using unselective indexes and there for reading more index blocks than necessary or using the wrong indexes.  So these waits may point to poor execution plans for SQL(s).    In either case, the next step would be to check the top resource consuming SQL(s) from the AWR report to determine whether these look excessive or whether improvements can be made.   To do this look at the [SQL Statistics](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=lnpz2w0u5_4&_afrLoop=269774337128792 \\l SQLStats) Section.   As mentioned, 20% of the time is spent waiting for or using CPU time. This should also be looked at when looking at the SQL Statistics.  Remember that the next step to take following the Top 5 Waits is dependent upon the findings within that section. In the example above, 3 of the waits point towards potentially Sub-optimal SQL so that should be the section investigated next.   Equally, if you do not see any latch waits, then latches are not causing a significant problem on your instance and so you do not need to investigate latch waits further.   Generally, if the database is slow, and the Top 5 timed events include "CPU" and "db file sequential read" and "db file scattered read" in any order, then it is usually worth jumping to the Top SQL (by logical and physical reads) section of an AWR report and calling the SQL Tuning Advisor on them (or tune them manually) just to make sure that they are running efficiently.   * **SQL Statistics**   AWR Reports show a number of different SQL statistics:  IMG_263  The different SQL statistic sub sections should be examined based upon the Top Wait events seen in the Top 5 Section.    In our example, we saw top waits as 'db file scattered read' , 'db file sequential read' and CPU. For these, we are most interested in  SQL ordered by CPU Time, Gets and Reads.  These sections actually duplicate some information adding other specifics as appropriate to the topic.   Often looking at 'SQL ordered by gets' is a convenient stating point as statements with high buffer gets are usually good candidates for tuning :  SQL ordered by Gets  -> Resources reported for PL/SQL code includes the resources used by all SQL    statements called by the code.  -> Total Buffer Gets:   4,745,943,815  -> Captured SQL account for     122.2% of Total                                 Gets              CPU     Elapsed   Buffer Gets   Executions    per Exec   %Total Time (s)  Time (s)    SQL Id  -------------- ------------ ------------ ------ -------- --------- -------------  1,228,753,877          168  7,314,011.2   25.9  8022.46   8404.73 5t1y1nvmwp2  SELECT ADDRESSID",CURRENT$."ADDRESSTYPEID",CURRENT$URRENT$."ADDRESS3",  CURRENT$."CITY",CURRENT$."ZIP",CURRENT$."STATE",CURRENT$."PHONECOUNTRYCODE",  CURRENT$."PHONENUMBER",CURRENT$."PHONEEXTENSION",CURRENT$."FAXCOU  1,039,875,759   62,959,363         16.5   21.9  5320.27   5618.96 grr4mg7ms81  Module: DBMS\_SCHEDULER  INSERT INTO "ADDRESS\_RDONLY" ("ADDRESSID","ADDRESSTYPEID","CUSTOMERID","  ADDRESS1","ADDRESS2","ADDRESS3","CITY","ZIP","STATE","PHONECOUNTRYCODE","PHONENU     854,035,223          168  5,083,543.0   18.0  5713.50   7458.95 4at7cbx8hnz  SELECT "CUSTOMERID",CURRENT$."ISACTIVE",CURRENT$."FIRSTNAME",CURRENT$."LASTNAME",CU<  RRENT$."ORGANIZATION",CURRENT$."DATEREGISTERED",CURRENT$."CUSTOMERSTATUSID",CURR  ENT$."LASTMODIFIEDDATE",CURRENT$."SOURCE",CURRENT$."EMPLOYEEDEPT",CURRENT$.  Tuning can either be performed either manually or by calling the SQL Tuning Advisor on them:  [Document 271196.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=271196.1) Automatic SQL Tuning - SQL Profiles.  [Document 262687.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=262687.1) How to use the Sql Tuning Advisor. [Document 276103.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=276103.1) PERFORMANCE TUNING USING ADVISORS AND MANAGEABILITY FEATURES: AWR, ASH, and ADDM and Sql Tuning Advisor.  NOTE: Use of the SQL Tuning Advisor requires the Oracle Tuning Pack License: [http://docs.oracle.com/database/121/DBLIC/options.htm#DBLIC139](http://docs.oracle.com/database/121/DBLIC/options.htm /l DBLIC139 /o Feature Availability by Edition)  ***Analysis:***     * -> Total Buffer Gets: 4,745,943,815  On the assumption that this is an hour long report, this is a significant number of gets and as such this confirms that it is worth investigating the top SQL statements to make sure they are taking optimal paths. * Individual Buffer Gets  The buffer gets for the individual statements shown are very high with the lowest being 850 Million. These 3 statements actually point towards 2 different reasons for the large number of buffers:      * Excessive Buffer Gets/Execution  SQL\_IDs '5t1y1nvmwp2' and '4at7cbx8hnz' are only executed 168 times, but each execution reads over 5 Million buffers. This SQL statement is a prime candidate for tuning since the number of buffers read in each execution is so high. * Excessive Executions  On the other hand SQL\_ID 'grr4mg7ms81' only reads 16 buffers for each execution. Tuning the individual statement may not be able to reduce that significantly. However, the issue with this statement is caused by the number of times it is executed - 65 Million.  Changing the way in which the statement is called is likely to have the largest impact here - it is likely that the statement is called in a loop, once per record, if it could be called so as to process multiple records at once then there is potential for significant economies of scale.   Remember that these numbers may be 'normal' for this environment (since some are very busy).  By comparing this report against a baseline, you can see whether these SQL statements also read this much data when the database performs well. If they do then they are not the cause of the issue and can be ignored (although there may be benefit generally in improving them).  ***Other SQL Statistic Sections***  As mentioned previously, there are a number of different report sections that help for specific causes. If you do not have the particular cause, then there is likely to be little benefit in looking at these. The following section outlines some potential causes and uses:   * **Load Profile**   Dependent on the waits, the load profile section either provides useful general background information or specific details related to potential issues.  Load Profile  ~~~~~~~~~~~~                            Per Second       Per Transaction                                    ---------------       ---------------                   Redo size:          4,585,414.80          3,165,883.14               Logical reads:             94,185.63             65,028.07               Block changes:             40,028.57             27,636.71              Physical reads:              2,206.12              1,523.16             Physical writes:              3,939.97              2,720.25                  User calls:                 50.08                 34.58                      Parses:                 26.96                 18.61                 Hard parses:                  1.49                  1.03                       Sorts:                 18.36                 12.68                      Logons:                  0.13                  0.09                    Executes:              4,925.89              3,400.96                Transactions:                  1.45   % Blocks changed per Read:   42.50    Recursive Call %:    99.19  Rollback per transaction %:   59.69       Rows per Sort:  1922.64  In the example, the waits section shows potential for issues with the execution of SQL so the load profile can be checked for details in this area, although it is not the primary source of such information.   If you were looking at the AWR report for general tuning, you might pick up that the load section shows relatively high redo activity with high physical writes. There are more writes than reads on this load with 42% block changes.    Furthermore, there is less hard parsing compared the soft parses.  If there was a mutex wait as top wait such as 'library cache: mutex X', then statistics such as the overall parse rate would be more relevant.    Again, comparing to a baseline will provide the best information, for example, checking to see if the load has changed by comparing redo size, users calls, and parsing.   * **Instance Efficiency**   Again, instance efficiency stats are more use for general tuning as opposed to addressing specific issues (unless waits point at these).  Instance Efficiency Percentages (Target 100%)  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~             Buffer Nowait %:   99.91       Redo NoWait %:  100.00             Buffer  Hit   %:   98.14    In-memory Sort %:   99.98             Library Hit   %:   99.91        Soft Parse %:   94.48          Execute to Parse %:   99.45         Latch Hit %:   99.97  Parse CPU to Parse Elapsd %:   71.23     % Non-Parse CPU:   99.00  The most important Statistic presented here from the point of view of our example is the '% Non-Parse CPU' because this indicates that almost all the CPU  time that we see in the Top Waits section is attributable to Execution and not parse, which means that tuning SQL may help to improve this.   If we were tuning, then 94.48% soft parse rate would show a small proportion of hard parsing which is desirable.  The high execute to parse % indicates good usage of cursors.  Generally, we want the statistics here close to 100%, but remember that a few percent may not be relevant dependent on the application.  For example, in a data warehouse environment, hard parsing may be higher due to usage of materialized views and, or histograms.  So again comparing to baseline report when performance was good is important.   * **Latch Activity**   In the example, we are not seeing significant waits for latches so this section could be ignored.   However, if latch waits were significant, then we would be looking for high latch sleeps under Latch Sleep Breakdown for latch free waits:  Latch Sleep Breakdown     \* ordered by misses desc  Latch Name  ----------------------------------------   Get Requests      Misses      Sleeps  Spin Gets   Sleep1   Sleep2   Sleep3  -------------- ----------- ----------- ---------- -------- -------- --------  cache buffers chains  2,881,936,948     3,070,271      41,336  3,031,456        0        0        0  row cache objects    941,375,571   1,215,395         852  1,214,606        0        0        0  object queue header operation    763,607,977     949,376      30,484    919,782        0        0        0  cache buffers lru chain    376,874,990     705,162       3,192    702,090        0        0        0  Here the top latch is cache buffers chains. Cache Buffers Chains latches protect the buffers in the buffer cache that hold data that we have retrieved from disk. This is a perfectly normal latch to see when data is being read. When this becomes stressed, the sleeps figure tends to rise as sessions start to wait to get the buffers they require. Contention can be caused by poorly tuned SQL reading the same buffers.   In our example, although the gets are high at 2.8 billion buffer gets, the sleeps at 41,336 is low.  Average number of sleeps per miss ratio (Avg Slps/Miss) is low. The reason for this is that the server is able to deal with this volume of data and so there is no significant contention on Cache Buffers Chains latches at this point.   For other latch free waits, review the following Document to identify what type of latches to investigate:  [Document 413942.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=413942.1) How to Identify Which Latch is Associated with a "latch free" wait  **Notable timed and wait events:**   * ***CPU time events***   Just because CPU comes as top timed event in AWR may not indicate a problem.  However, if performance is slow with high CPU usage, then start investigating the wait.  First, check to see if a sql is taking most CPU under SQL ordered by CPU Time in AWR:  SQL ordered by CPU Time  -> Resources reported for PL/SQL code includes the resources used by all SQL    statements called by the code.  -> % Total is the CPU Time divided into the Total CPU Time times 100  -> Total CPU Time (s):          56,207  -> Captured SQL account for      114.6% of Total     CPU      Elapsed                  CPU per          % Total   Time (s)   Time (s)  Executions     Exec (s) % Total DB Time SQL Id  ---------- ---------- ------------ ----------- ------- ------- -------------      20,349     24,884          168      121.12    36.2     9.1 7bbhgqykv3cm9  Module: DBMS\_SCHEDULER  DECLARE job BINARY\_INTEGER := :job; next\_date TIMESTAMP WITH TIME ZONE := :myda  te; broken BOOLEAN := FALSE; job\_name VARCHAR2(30) := :job\_name; job\_subname  VARCHAR2(30) := :job\_subname; job\_owner VARCHAR2(30) := :job\_owner; job\_start  TIMESTAMP WITH TIME ZONE := :job\_start; job\_scheduled\_start TIMESTAMP WITH TIME  ***Analysis:***     * -> Total CPU Time (s): 56,207  This represents 15 minutes of CPU time in total. Whether this is significant depends on the report duration. * The top CPU using SQL uses 20,349 second (around 5 minutes), * Total DB of time this represents is 9.1%. * Executions is 168 - being as this execution count is the same as 2 of the 3 SQLs identified earlier, these may be related and this task may well be the scheduling job that runs the SQLs.   ***Actions:***  Once you have identified the SQL statements that are using the highest CPU, investigate the reason for this usage.     * Look at the number of executions and see whether that is appropriate for this statement. Excessive executions might indicate that the statement is being called too frequently and it might be possible to execute it for a group of rows rather than row by row (i.e. execute it in a batch). * Is the amount of CPU per execution excessive - this might indicate that the statement itself is inefficient. * Additionally, look at the other SQL Statistics in the AWR report to see if the SQLID(s) in question show excessive values for any of those, then deal with the statement appropriately.   ***Other Potential CPU related Issues:***     * ***Check to see if other waits follow the high CPU timed event.***   For example, cursor: pin S waits may cause the high CPU with following known issue:  [Document 6904068.8](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=6904068.8) Bug 6904068 - High CPU usage when there are "cursor: pin S" waits     * ***High External CPU usage***   If a process outside of the database is taking high CPU, then this could be preventing database processes from getting the CPU they require and affecting the database performance. In this case, run oswatcher or other os diagnostic tools to find which process is taking high CPU.  [Document 433472.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=433472.1) OS Watcher For Windows (OSWFW) User Guide     * ***Troubleshooting CPU usage***   The following Document outlines how to further diagnose high CPU usage:  [Document 164768.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=164768.1) Troubleshooting: High CPU Utilization   * ***'Log file sync' waits***   When a user session commits or rolls back, the log writer flushes the redo from log buffer to the redo logs. AWR reports are very useful for determination if this is a problem and whether the cause of the problem is I/O or in some other area. The following articles deal specifically with this symptom:  [Document 1376916.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1376916.1) Troubleshooting: "Log File Sync" Waits  [Document 34592.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=34592.1)WAITEVENT: "log file sync"   * ***Buffer busy waits***   This is he event waited on when a session is trying to get a buffer from the buffer cache but the buffer is busy - either being read by another session or another session is holding it in incompatible mode.  In order to find which block is busy and why, use the following Documents:  [Document 155971.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=155971.1) Resolving Intense and "Random" Buffer Busy Wait Performance Problems [Document 34405.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=34405.1) WAITEVENT: "buffer busy waits"   * ***Waits for 'Cursor: mutex/pin'***   If there are mutex waits such such as 'Cursor: pin S wait on X' or 'Cursor: mutex X' etc , then these are indicative of parsing issues. On this basis look for statements with high parse counts or high version counts under 'SQL ordered by Parse Calls' and 'SQL ordered by Version Count' as these are most likely to be the causes of problems. The following Documents can assist further:  [Document 1356828.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1356828.1) FAQ: 'cursor: mutex ..' / 'cursor: pin ..' / 'library cache: mutex ..' Type Wait Events  [Document 1349387.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1349387.1) Troubleshooting 'cursor: pin S wait on X' waits.    **Troubleshooting Other Issues**  For guidance troubleshooting other performance issues see:  [Document 1377446.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1377446.1) Troubleshooting Performance Issues  **Use of ADDM Reports alongside AWR**  ADDM reports can be reviewed along with AWR to assist in diagnosis since they provide specific recommendations which can help point at potential problems. The following is a sample ADDM report taken from:  [Document 250655.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=250655.1)How to use the Automatic Database Diagnostic Monitor:  Example Output:  DETAILED ADDM REPORT FOR TASK 'SCOTT\_ADDM' WITH ID 5  ----------------------------------------------------   Analysis Period: 17-NOV-2003 from 09:50:21 to 10:35:47  Database ID/Instance: 494687018/1  Snapshot Range: from 1 to 3  Database Time: 4215 seconds  Average Database Load: 1.5 active sessions   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~    FINDING 1: 65% impact (2734 seconds)  ------------------------------------  PL/SQL execution consumed significant database time.   RECOMMENDATION 1: SQL Tuning, 65% benefit (2734 seconds)  ACTION: Tune the PL/SQL block with SQL\_ID fjxa1vp3yhtmr. Refer to  the "Tuning PL/SQL Applications" chapter of Oracle's "PL/SQL  User's Guide and Reference"  RELEVANT OBJECT: SQL statement with SQL\_ID fjxa1vp3yhtmr  BEGIN EMD\_NOTIFICATION.QUEUE\_READY(:1, :2, :3); END;   FINDING 2: 35% impact (1456 seconds)  ------------------------------------  SQL statements consuming significant database time were found.   RECOMMENDATION 1: SQL Tuning, 35% benefit (1456 seconds)  ACTION: Run SQL Tuning Advisor on the SQL statement with SQL\_ID  gt9ahqgd5fmm2.  RELEVANT OBJECT: SQL statement with SQL\_ID gt9ahqgd5fmm2 and  PLAN\_HASH 547793521  UPDATE bigemp SET empno = ROWNUM   FINDING 3: 20% impact (836 seconds)  -----------------------------------  The throughput of the I/O subsystem was significantly lower than expected.   RECOMMENDATION 1: Host Configuration, 20% benefit (836 seconds)  ACTION: Consider increasing the throughput of the I/O subsystem.  Oracle's recommended solution is to stripe all data file using  the SAME methodology. You might also need to increase the  number of disks for better performance.   RECOMMENDATION 2: Host Configuration, 14% benefit (584 seconds)  ACTION: The performance of file  D:\ORACLE\ORADATA\V1010\UNDOTBS01.DBF was significantly worse  than other files. If striping all files using the SAME  methodology is not possible, consider striping this file over  multiple disks.  RELEVANT OBJECT: database file  "D:\ORACLE\ORADATA\V1010\UNDOTBS01.DBF"   SYMPTOMS THAT LED TO THE FINDING:  Wait class "User I/O" was consuming significant database time.  (34% impact [1450 seconds])   FINDING 4: 11% impact (447 seconds)  -----------------------------------  Undo I/O was a significant portion (33%) of the total database I/O.   NO RECOMMENDATIONS AVAILABLE   SYMPTOMS THAT LED TO THE FINDING:  The throughput of the I/O subsystem was significantly lower than  expected. (20% impact [836 seconds])  Wait class "User I/O" was consuming significant database time.  (34% impact [1450 seconds])   FINDING 5: 9.9% impact (416 seconds)  ------------------------------------  Buffer cache writes due to small log files were consuming significant  database time.   RECOMMENDATION 1: DB Configuration, 9.9% benefit (416 seconds)  ACTION: Increase the size of the log files to 796 M to hold at  least 20 minutes of redo information.  ADDM report gives possible recommendations in more readable format than AWR.  However, ADDM should be interpreted along with AWR statistics for accurate diagnostics.  **Other AWR reference Articles**  The following documents can assist when reading other sections of AWR reports and for other purposed:  Document 786554.1 How to Read PGA Memory Advisory Section in AWR and Statspack Reports [Document 754639.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=754639.1) How to Read Buffer Cache Advisory Section in AWR and Statspack Reports  [Document 1301503.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1301503.1) Troubleshooting: AWR Snapshot Collection issues  [Document 1363422.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=1363422.1) Automatic Workload Repository (AWR) Reports - Start Point  **Statspack**  AWR reports supersede legacy reports such as statspack and bstat/estat. For reference, the following is a link to and article outlining how to read statspack reports:  [http://www.oracle.com/technetwork/database/focus-areas/performance/statspack-opm4-134117.pdf](http://www.oracle.com/technetwork/database/focus-areas/performance/statspack-opm4-134117.pdf /o How to Interpret Statspack reports)  Additional information can be found in the following articles:  Document 94224.1 FAQ- Statspack Complete Reference [Document  394937.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=394937.1) Statistics Package (STATSPACK) Guide   [Document 149113.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=149113.1)  Installing and Configuring StatsPack Package [Document 149121.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=149121.1) Gathering a StatsPack snapshot [Document 228913.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1359094.1&id=228913.1)  Systemwide Tuning using STATSPACK Reports |