

General Understanding of AWR Reports

1. Q: What is an AWR report, and how is it used in performance tuning?

A: The Automatic Workload Repository (AWR) is a built-in repository in Oracle databases that collects, processes, and maintains performance statistics. It captures data on system load, wait events, SQL execution, and more at regular intervals (snapshots). AWR reports summarize this data over a specified time period, helping DBAs analyze performance, identify bottlenecks, and tune the database.

2. Q: How do you generate an AWR report for a specific time interval?

A: You can generate an AWR report using the `awrrpt.sql` script found in the `$ORACLE_HOME/rdbms/admin` directory. Run the script in SQL*Plus, and it will prompt you to select the database ID, instance number, and snapshot IDs corresponding to the desired time interval.

3. Q: What are snapshots in the context of AWR, and how frequently are they taken by default?

A: Snapshots are periodic data collections of database performance metrics stored in the AWR. By default, Oracle takes snapshots every hour and retains them for 8 days. The frequency and retention period can be adjusted using `DBMS_WORKLOAD_REPOSITORY` package procedures.

SQL Tuning

4. Q: In the AWR report, how can you identify SQL statements that are consuming the most resources?

A: Look under the "SQL Statistics" sections, such as "SQL Ordered by Elapsed Time," "SQL Ordered by CPU Time," or "SQL Ordered by Gets." These sections list SQL statements sorted by resource consumption, helping identify those that are heavy hitters.

5. Q: What is the significance of the "Elapsed Time" metric in SQL ordered sections of the AWR report?

A: "Elapsed Time" represents the total time taken by a SQL statement during the snapshot interval. It includes CPU time, wait times, and any other overheads. High elapsed times can indicate inefficient queries that may need tuning.

6. Q: How can high "Buffer Gets" in a SQL statement indicate a performance issue?

A: "Buffer Gets" refers to the number of logical I/O operations performed. A high number of buffer gets can indicate that a SQL statement is scanning large amounts of data, possibly due to missing indexes or suboptimal execution plans, leading to increased CPU usage.

7. Q: Explain how the "Execution Plan" of a SQL statement affects its performance.

A: The execution plan outlines how Oracle retrieves data for a SQL statement. It details the operations (e.g., table scans, index scans, joins) and their order. A suboptimal execution plan may result in excessive I/O, CPU usage, and longer execution times.

8. Q: What steps can you take if a SQL statement shows a high "Executions" count with low "Elapsed Time" per execution?

A: While each execution may be fast, the cumulative effect can impact performance. Consider batching operations, reducing frequency, or caching results if appropriate. Also, verify whether the application logic can be optimized to minimize unnecessary executions.

9. Q: How does the use of bind variables influence SQL performance as seen in the AWR report?

A: Bind variables promote cursor sharing, reducing hard parses and improving performance. The AWR report's "Instance Efficiency Percentages" section shows the "Parse CPU to Parse Elapsed %" and "Soft Parse %" metrics, indicating parsing efficiency.

10. Q: What does a high number of "Physical Reads" in a SQL statement indicate?

A: High "Physical Reads" suggest that the SQL statement is causing disk I/O operations, which are slower than memory operations. This could be due to full table scans, lack of indexing, or insufficient buffering.

Batch Jobs Analysis

11.Q: How can AWR reports help in analyzing the performance impact of batch jobs?

A: By generating an AWR report covering the batch job's execution period, you can examine system load, wait events, and resource consumption. Sections like "Load Profile" and "Top SQL" help identify resource-intensive operations associated with the batch job.

12.Q: What indicators in the AWR report suggest that a batch job is causing contention?

A: High wait events such as "enq: TX - row lock contention" or spikes in CPU and I/O usage during the batch window can indicate that the batch job is causing contention with other database activities.

13.Q: How can you use AWR reports to optimize batch job scheduling?

A: By analyzing workload patterns in AWR reports, you can identify periods of low system utilization and schedule batch jobs during these times to minimize impact on interactive users.

14.Q: What AWR metrics would you examine to assess the I/O impact of a batch job?

A: Review "Physical Read Total Bytes," "Physical Write Total Bytes," and wait events related to I/O, such as "db file sequential read" and "db file scattered read," to assess the batch job's I/O footprint.

15.Q: How can excessive PGA usage during batch jobs be detected in an AWR report?

A: The "PGA Aggr Target Stats" and "PGA Memory Advisory" sections show PGA usage. High values of "Total PGA Allocated" and significant "PGA Memory Over-Allocations" indicate excessive PGA consumption.

PGA (Program Global Area) and SGA (System Global Area)

16.Q: What is the difference between PGA and SGA in Oracle databases?

A: The PGA is memory allocated per server process for operations like sorting and hashing. The SGA is shared memory used by all server processes, containing structures like the buffer cache, shared pool, and redo log buffer.

17.Q: How does the "PGA Aggregate Target" setting affect database performance?

A: It defines the target aggregate PGA memory available to all server processes. Proper sizing ensures efficient sorting and hashing operations, reducing disk I/O and improving query performance.

18.Q: In an AWR report, where can you find information about PGA memory usage?

A: The "Memory Statistics" section, specifically under "PGA Aggr Target Stats," provides insights into PGA memory usage, including total PGA allocated and in-use statistics.

19.Q: What does a high "PGA Memory Over-Allocation Count" indicate?

A: It suggests that the PGA usage has exceeded the PGA aggregate target multiple times, leading to potential performance issues due to insufficient memory for optimal processing.

20.Q: How can you determine if the SGA size is appropriate using an AWR report?

A: Check the "Instance Efficiency Percentages" and "Memory Statistics" sections. High values in "Buffer Cache Hit Ratio" and "Library Cache Hit Ratio" generally indicate efficient SGA sizing.

21.Q: What is the impact of an undersized shared pool in the SGA?

A: An undersized shared pool can lead to increased hard parses, library cache misses, and fragmentation, resulting in higher CPU usage and slower query performance.

22.Q: How can you detect shared pool contention in an AWR report?

A: Look for wait events like "library cache lock" or "library cache pin" and high values in "parse time elapsed" relative to "parse CPU time" in the "Time Model Statistics" section.

23.Q: What role does the buffer cache play in database performance?

A: The buffer cache stores copies of data blocks read from disk. A properly sized buffer cache reduces physical I/O by satisfying requests from memory, thus improving performance.

24.Q: How can you assess buffer cache efficiency in an AWR report?

A: Examine the "Buffer Cache Hit Ratio" in the "Instance Efficiency Percentages" section. Ratios close to 100% indicate efficient use, but excessively high values with performance issues may suggest other problems.

25.Q: What is the "Redo Log Buffer," and how does its size affect performance?

A: The redo log buffer caches redo entries before writing to disk. If it's too small, processes may wait for space, leading to "log buffer space" waits. Proper sizing reduces such waits and improves transaction throughput.

Locks and Contention

26.Q: How can lock contention be identified in an AWR report?

A: High wait events like "enq: TX - row lock contention" or "enq: TM - contention" indicate lock waits. The "Segments by Row Lock Waits" section shows which segments are involved.

27.Q: What does the wait event "enq: TX - row lock contention" signify?

A: It indicates that a session is waiting for a row-level lock held by another transaction, often due to uncommitted changes or long-running transactions holding locks.

28.Q: How can you determine which sessions or transactions are causing lock contention?

A: Use the "Active Session History" (ASH) data in the AWR report, focusing on sessions with high wait times for lock-related events. Cross-reference with the "Blocking Sessions" section if available.

29.Q: What strategies can be employed to reduce lock contention identified in an AWR report?

A: Optimize application logic to commit transactions promptly, avoid unnecessary locks (e.g., SELECT FOR UPDATE when not needed), and consider using row-level locking appropriately.

30.Q: How does indexing influence lock contention?

A: Proper indexing can reduce the number of rows locked during DML operations by narrowing down the affected rows, thus minimizing contention.

Buffers and I/O Wait Events

31.Q: What is the difference between "db file sequential read" and "db file scattered read" wait events?

A: "db file sequential read" indicates single-block reads, often associated with index lookups. "db file scattered read" involves multi-block reads, typically from full table scans or index fast full scans.

32.Q: In an AWR report, how would high "db file sequential read" waits be interpreted?

A: It suggests significant time spent on single-block I/O operations, possibly due to inefficient index usage or excessive index range scans.

33.Q: What might cause high "db file scattered read" wait times?

A: This indicates heavy multi-block reads, often due to full table scans. It could be caused by missing indexes, large table sizes, or inefficient query plans.

34.Q: How can you use the AWR report to determine if the database is experiencing I/O bottlenecks?

A: High wait times for I/O-related events (e.g., "db file sequential read") and high average wait times indicate potential I/O bottlenecks. Additionally, the "IO Profile" section provides I/O throughput metrics.

35.Q: What is the significance of the "Buffer Busy Waits" event?

A: It occurs when multiple sessions attempt to access the same block in the buffer cache, but the block is currently being read or modified by another session. High occurrences suggest contention in the buffer cache.

36.Q: How can you reduce "Buffer Busy Waits" identified in the AWR report?

A: Options include increasing the buffer cache size, optimizing application code to reduce contention on hot blocks, and examining segment-level statistics to identify problematic objects.

Wait Events Analysis

37.Q: What are wait events in Oracle, and why are they important for performance analysis?

A: Wait events represent the time sessions spend waiting for resources or conditions. Analyzing wait events helps identify bottlenecks and areas where the database is not efficiently utilizing resources.

38.Q: How do you prioritize which wait events to address from an AWR report?

A: Focus on wait events with the highest "Total Wait Time" or "Wait Class" percentages. Addressing the top wait events typically yields the most significant performance improvements.

39.Q: What does a high "log file sync" wait event indicate?

A: It suggests that sessions are waiting for commits to complete, which involves writing redo data to disk. This could be due to slow I/O subsystem, frequent commits, or insufficient log buffer size.

40.Q: How can you reduce "log file sync" wait times?

A: Improve I/O performance of the redo logs (e.g., faster disks, SSDs), batch transactions to reduce commit frequency, or increase the size of the redo log buffer if waits are due to buffer space issues.

41.Q: Explain the "latch: cache buffers chains" wait event and its impact.

A: This wait event indicates contention for cache buffer chain latches, which protect hash chains of buffer headers in the buffer cache. High waits suggest hot blocks or heavy concurrent access to specific buffers.

42.Q: How can "latch: cache buffers chains" contention be alleviated?

A: Identify hot blocks using V\$BH and V\$CACHE, and spread out access by partitioning data, optimizing SQL to access data less frequently, or adding indexes to reduce unnecessary full scans.

43.Q: What is the significance of the "library cache lock" wait event?

A: It indicates contention in the library cache, often due to high parsing rates, invalidations, or concurrent DDL operations. This can lead to increased CPU usage and slower query performance.

44.Q: How can you address "library cache lock" waits?

A: Reduce hard parsing by using bind variables, minimize DDL operations during peak times, and ensure the shared pool is adequately sized.

Instance Efficiency Metrics

45.Q: What does the "Instance Efficiency Percentages" section of the AWR report represent?

A: It provides ratios that indicate how efficiently the database instance is operating, such as the buffer cache hit ratio, library cache hit ratio, and parse ratios.

46.Q: Is a high buffer cache hit ratio always indicative of good performance?

A: Not necessarily. While a high hit ratio means most data requests are served from memory, it doesn't account for inefficient queries or unnecessary data access patterns. It's important to look at other metrics as well.

47.Q: What does a low "Soft Parse %" in the AWR report suggest?

A: It indicates a high proportion of hard parses, which are resource intensive. This can be caused by not using bind variables, leading to increased CPU usage and potential contention in the shared pool.

48.Q: How can you improve the "Soft Parse %" ratio?

A: Use bind variables to promote cursor sharing, reduce the frequency of SQL statement parsing, and ensure that the shared pool is adequately sized to store execution plans.

49.Q: What does the "Execute to Parse %" metric indicate?

A: It measures the proportion of executions to parses. A high percentage suggests that SQL statements are being reused efficiently, reducing parsing overhead.

50.Q: How can a low "Latch Hit %" affect database performance?

A: It indicates contention for latches, which are lightweight synchronization mechanisms. Low latch hit ratios can lead to increased wait times and reduced throughput.

Database Performance Metrics

51.Q: What is the "Load Profile" section of the AWR report, and what information does it provide?

A: The "Load Profile" summarizes key workload characteristics during the snapshot interval, including transactions per second, logical and physical reads, redo size, and other metrics that help assess the database load.

52.Q: How can you use the "Transactions per Second" metric in performance analysis?

A: It indicates the transaction throughput of the database. Sudden changes may point to workload shifts or performance issues affecting transaction processing.

53.Q: What does a high "Redo Size per Transaction" imply?

A: It suggests that transactions are generating a large amount of redo data, possibly due to extensive DML operations or inefficient batch processing. This can impact redo log performance and storage requirements.

54.Q: How is "Logical Reads per Second" relevant to performance?

A: It represents the volume of memory reads from the buffer cache. High values may indicate intensive data processing workloads, necessitating analysis of SQL statements, and indexing strategies.

55.Q: What can you infer from the "User Calls per Second" metric?

A: It reflects the number of calls made by client sessions to the database. High values may be due to chatty applications making frequent calls, which can increase network overhead and processing load.

56.Q: How does the "Top Timed Events" section help in identifying bottlenecks?

A: It lists the wait events that consumed the most time during the snapshot period, allowing you to focus on the most significant performance inhibitors.

57.Q: What is the significance of the "Time Model Statistics" section?

A: It provides a breakdown of where the database spent time during processing, such as in parsing, execution, or commit operations, helping identify areas that may need optimization.

58.Q: How can the "Advisory Statistics" in the AWR report guide performance tuning?

A: Advisory sections like "Buffer Cache Advisory" and "PGA Memory Advisory" offer recommendations on memory sizing based on simulated workloads, helping you adjust settings for optimal performance.

59.Q: What does the "Wait Class" breakdown in the AWR report indicate?

A: It categorizes wait events into classes (e.g., User I/O, System I/O, Concurrency) and shows the proportion of total wait time for each class, aiding in identifying systemic issues.

60.Q: How can "OS Statistics" in the AWR report contribute to performance analysis?

A: Provides insights into system-level resource usage, such as CPU and memory utilization, which helps determine if performance issues are due to database configuration or underlying hardware constraints.

Advanced Analysis Techniques

61.Q: How can you correlate database CPU usage with SQL execution in the AWR report?

A: Examine the "SQL Ordered by CPU Time" section to identify SQL statements consuming the most CPU. Cross-reference with overall CPU usage statistics to assess their impact on database performance.

62.Q: What role does "ASH Data" play in enhancing AWR analysis?

A: Active Session History (ASH) samples session activity, providing granular insights into session behavior over time. ASH data can help pinpoint transient issues not apparent in aggregated AWR statistics.

63.Q: How can you use the "Segment Statistics" section to identify problematic database objects?

A: It lists segments with the highest activity in terms of logical reads, physical reads, or row lock waits, allowing you to focus tuning efforts on specific tables or indexes.

64.Q: What is the importance of monitoring "Undo Statistics" in the AWR report?

A: High undo generation can indicate excessive DML operations or long-running transactions, which may impact performance and require adjustments to undo tablespace sizing.

65.Q: How can you assess whether parallel execution is effectively utilized in the database?

A: Review the "Parallel Execution" statistics in the AWR report to see metrics like the number of parallel operations and the efficiency of parallel workers, helping determine if parallelism is benefiting performance.