

Performance Tuning Interview Questions

Question:: What are the major focuses of Performance tuning?

Performance tuning focuses primarily on writing efficient SQL, allocating appropriate computing resources, and analyzing wait events and contention in a system.

Question:: How does Oracle aid performance tuning?

Oracle provides several options to aid performance tuning, such as partitioning large tables, using materialized views, storing plan outlines, using tools like Automatic Optimizer statistics collection feature, packages like DBMS_STATS, SQL Tuning Advisor to tune SQL statements, etc.

Question:: Why is performance tuning a menacing area for DBA's?

Like many other features of Oracle like exp/imp, backup recovery this field can't be automated. This is one area that requires a lot of detective work on the part of application programmers and DBA's to see where some process is running slower than expected, why can't we scale applications to a larger number of users without problems like performance degradation etc. This is an area where our technical knowledge must be used along with constant experimentation and observation.

Question:: What are the approaches towards performance tuning?

We can follow either a systematic approach or a reactive approach for performance tuning.

What is a systematic approach to performance tuning?

It is mandatory to design the database properly at initial stages to avoid potential problems. It is mandatory to know the nature of application that a database is going to support. With a clear idea on the application's nature database can be optimally created by allocating appropriate resources to avoid problems when the application is moved to production. Most production moves cause problems because of the scalability problems with the applications. So, Oracle recommends to tune database at inception stage. This is systematic approach to performance tuning.

Question:: What are the Oracle's suggestions towards systematic tuning?

Oracle suggests a specific design approach with the following steps. This is a top down approach:

- 1) Design the application correctly
- 2) Tune the application SQL code
- 3) Tune memory
- 4) Tune I/O
- 5) Tune contention and other issues

Question:: What are the effects of poor database design?

A poor database design results in poor application performance. We have to tune the application code and some database resources such as memory, CPU, I/O owing to performance degradation. An application performs well in development and testing. Will there be any performance problem when it is moved to production?

Production moves may cause problems due to scalability. We can't simulate the original load in

test and development. So problems may crop up at times as the application may be performing poor due to scalability problems.

Question:: What is reactive performance tuning?

Performance tuning is an iterative process. We as a DBA may have to tune applications which is designed and implemented in production. The performance tuning at this stage is referred to as reactive performance tuning.

Question:: Which is useful – systematic or reactive tuning?

The performance tuning steps to improve the performance of a database depends on the stage at which we get the input and on the nature of the application. DBA's can assist the developers to write optimal code that is scalable based on systematic approach. Mostly the real life problems that are encountered after production moves have to be solved by reactive performance tuning.

Question:: We have an application whose code can't be changed. Can we improve its performance?

We can improve the application performance without changing base SQL code by optimizing the SQL performance. Oracle has come up with SQL Advisor tool that helps SQL performance. We can make use of SQL Advisor tools' SQL Profiles to improve performance, though we can't touch the underlying SQL.

Question:: What is the use of SQL over procedural languages?

SQL isn't a procedural language in which we have to specify the steps to be followed to achieve the statement goal. We don't have to specify how to accomplish a task (say data retrieval) using SQL, rather we can specify as to what needs to be done.

Question:: What is query processing?

When a user starts a data retrieval operation, the user's SQL statement goes through several sequential steps that together constitute query processing. Query processing is the transformation of the SQL statement into efficient execution plan to return the requested data from the database.

Question:: What is query optimization?

Query optimization is the process of choosing the most efficient execution plan. The goal is to achieve the result with least cost in terms of resource usage. Resources include I/O and CPU usage on the server where the database is running. This is a means to reduce the execution times of the query, which is the sum of the execution times of the all component operations of the query.

Question:: What are the techniques used for query optimization?

Cost-based optimization, heuristic strategy are used for query optimization.

Question:: What are the phases of a SQL statement processing?

An user's SQL statement goes through the parsing, optimizing, and execution stages. If the SQL statement is a query (SELECT), data has to be retrieved so there's an additional fetch stage before the SQL processing is complete.

Question:: What is Parsing?

Parsing primarily consists of checking the syntax and semantics of the SQL statements. The end product of the parse stage of query compilation is the creation of a parse tree, which represents the query structure. The parse tree is then sent to the logical query plan generation stage.

Mention the steps in the creation of a parse tree:-

- 1) The SQL statement is decomposed into relational algebra query that 's analyzed to see whether it's syntactically correct.
- 2) The query then undergoes semantic checking.
- 3) The data dictionary is consulted to ensure that the tables and the individual columns that are referenced in the query do exist, as well as all the object privileges.
- 4) The column types are checked to ensure that the data matches the column definition.
- 5) The statement is normalized so that it can be processed more efficiently
- 6) The query is rejected if it is incorrectly formulated
- 7) Once the parse tree passes all the syntactic and semantic checks, it is considered a valid parse tree, and it's sent to the logical query plan generation stage.

Question:: Where does the parse tree generation take place?

The parse tree generation takes place in the library cache portion of the SGA(system global Area)./

What is Optimization/what happens during optimization phase?

During the optimization phase, Oracle uses its optimizer(CBO(cost-based optimizer)) to choose the best access method for retrieving data for the tables and indexes referred to in the query.

Question:: How does a CBO generate an optimal execution plan for the SQL statement?

Using the statistics we provide and the hints specified in the SQL queries, the CBO produces an optimal execution plan for the SQL statement.

What are the parts of an optimizer phase?

An optimizer phase can be divided into two distinct parts: the query rewrite phase and the physical execution plan generation phase.

Question:: What is query rewrite phase?

In this phase, the parse tree is converted into an abstract logical query plan. This is an initial pass at an actual query plan, and it contains only a general algebraic reformulation of the initial query. The various nodes and branches of the parse tree are replaced by operators of relational algebra.

Question:: What is execution plan generation phase/physical execution plan generation phase?

During this phase, Oracle transforms the logical query plan into a physical query plan.

The optimizer may be faced with a choice of several algorithms to solve a query. It needs to choose the most efficient algorithm to answer a query, and it needs to determine the most efficient way to implement the operations. The optimizer determines the order in which it will perform the steps.

Question:: What are the factors considered by a physical query/execution plan?

Following factors are considered by a physical query or an execution plan:

- 1) The various operations(eg: joins) to be performed during the query

- 2) The order in which the operations are performed
- 3) The algorithm to be used for performing each operation
- 4) The best way to retrieve data from disk or memory
- 5) The best way to pass data from one operation to another during the query

Question:: Which generates the query plan/what is generated by optimizer?

The optimizer generates several valid physical query plans. All the physical query plans are potential execution plans.

Question:: How does the optimizer choose the query plan/what is cost-based query optimization?/

The optimizer generates several physical query plans that are potential execution plans. The optimizer then chooses among them by estimating the cost of each possible physical plan based on the table and index statistics available to it, and selecting the plan with the lowest estimated cost. This evaluation of the possible physical query plans is called cost-based query optimization.

Question:: What are the factors affecting the cost of a execution plan?

The cost of executing a plan is directly proportional to the amount of resources such as I/O, memory and CPU necessary to execute the proposed plan.

Question:: What happens after choosing the low-cost physical query plan?

The optimizer passes the low-cost physical query plan to the Oracle's query execution engine.

Question:: What is a heuristic strategy?

The database uses a less systematic query optimization technique known as the heuristic strategy.

Question:: What are unary and binary operations?

A join operation is called a binary operation, an operation like selection is called a unary operation.

Question:: What is an optimal operation processing strategy?

In general an optimal strategy is to perform unary operations first so the more complex and time-consuming binary operations use smaller operands. Performing as many of the possible unary operations first reduces the row sources of the join operations.

Question:: What are the heuristic-processing strategies?

- 1) Perform selection operation early so that we can eliminate a majority of the candidate rows early in the operation. If we leave most rows in until the end, we're going to do needless comparisons with the rows we're going to get rid of later
- 2) Perform projection operations early so that we limit the number of columns we have to deal with
- 3) If we need to perform consecutive join operation, perform the operations that produce the smaller join first
- 4) Compute common expressions once and save the results

Question:: What is query execution?

During the final stage of a query processing, the optimized query(the physical query plan that has been selected) is executed. If it's a SELECT statement the rows are returned to the user.If it's an INSERT,UPDATE or DELETE statement ,the rows are modified. The SQL execution engine takes the execution plan provided by the optimization phase and executes it.

Question:: What is the crucial step in SQL statement processing?

Of the three steps involved in the SQL statement processing, the optimization process is the crucial one because it determines the all important question of how fast our data will be retrieved./

Question:: What is the job of an optimizer?

The job of an optimizer is to find the optimal/best plan to execute our DML statements such as SELECT,INSERT,UPDATE and DELETE.Oracle uses CBO to help determine efficient methods to execute queries.

Question:: What is an index?

An index is a data structure that takes the value of one or more columns of a table(the key) and returns all rows/requested-columns in a row quickly.

Question:: Why is an index efficient?

The efficiency of an index comes from the fact that it lets us find necessary rows without having to scan all the rows of a table.They need a fewer disk I/O's than if we had to scan the table and hence are efficient.

Question:: When do we need to index tables?

We need to index tables only when the queries will be selecting a small portion of the table.If our query is retrieving rows that are greater than 10 or 15 percent of the total rows in the table,we may not need an index.

Question:: Why does an index traverses a table's row faster?

Indexes prevent a full table scan,so it is inherently a faster means to traverse a table's row

Question:: A tablespace has a table with 30 extents in it. Is this bad? Why or why not?

Multiple extents in and of themselves aren't bad. However if you also have chained rows this can hurt performance.

Question::How do you set up tablespaces during an Oracle installation?

You should always attempt to use the Oracle Flexible Architecture standard or another partitioning scheme to ensure proper separation of SYSTEM, ROLLBACK, REDO LOG, DATA, TEMPORARY and INDEX segments.

Question:: You see multiple fragments in the SYSTEM tablespace, what should you check first? Ensure that users don't have the SYSTEM tablespace as their TEMPORARY or DEFAULT tablespace assignment by checking the DBA_USERS view.

Question:: What are some indications that you need to increase the SHARED_POOL_SIZE parameter?

Poor data dictionary or library cache hit ratios, getting error ORA-04031. Another indication is steadily decreasing performance with all other tuning parameters the same.

Question:: What is the general guideline for sizing db_block_size and db_multi_block_read for an application that does many full table scans?

Oracle almost always reads in 64k chunks. The two should have a product equal to 64 or a multiple of 64.

Question:: What is the fastest query method for a table?

Fetch by rowid

Question:: Explain the use of TKPROF? What initialization parameter should be turned on to get full TKPROF output?

The tkprof tool is a tuning tool used to determine cpu and execution times for SQL statements. You use it by first setting timed_statistics to true in the initialization file and then turning on tracing for either the entire database via the sql_trace parameter or for the session using the ALTER SESSION command. Once the trace file is generated you run the tkprof tool against the trace file and then look at the output from the tkprof tool. This can also be used to generate explain plan output.

Question:: When looking at v\$sysstat you see that sorts (disk) is high. Is this bad or good? If bad, how do you correct it?

If you get excessive disk sorts this is bad. This indicates you need to tune the sort area parameters in the initialization files. The major sort area parameter is the SORT_AREA_SIZE parameter.

Question:: When should you increase copy latches? What parameters control copy latches?

When you get excessive contention for the copy latches as shown by the "redo copy" latch hit ratio. You can increase copy latches via the initialization parameter LOG_SIMULTANEOUS_COPIES to twice the number of CPUs on your system.

Question:: Where can you get a list of all initialization parameters for your instance? How about an indication if they are default settings or have been changed?

You can look in the init.ora file for an indication of manually set parameters. For all parameters, their value and whether or not the current value is the default value, look in the v\$parameter view.

Question:: Describe hit ratio as it pertains to the database buffers. What is the difference between instantaneous and total hit ratio; which should be used for tuning?

Hit ratio is a measure of how many times the database was able to read a value from the buffers versus how many times it had to re-read a data value from the disks. A value greater than 80-90% is good, less could indicate problems. If you take the ratio of existing parameters this will be a cumulative value since the database started. If you do a comparison between pairs of readings based on some arbitrary time span, this is the instantaneous ratio for that time span. Generally speaking an instantaneous reading gives more valuable data since it will tell you what your instance is doing for the time it was generated over.

Question:: Discuss row chaining, how does it happen? How can you reduce it? How do you correct it?

Row chaining occurs when a VARCHAR2 value is updated and the length of the new value is longer than the old value and won't fit in the remaining block space. This results in the row

chaining to another block. It can be reduced by setting the storage parameters on the table to appropriate values. It can be corrected by export and import of the effected table.

Question:: When looking at the estat events report you see that you are getting busy buffer waits. Is this bad? How can you find what is causing it?

Buffer busy waits may indicate contention in redo, rollback or data blocks. You need to check the v\$waitstat view to see what areas are causing the problem. The value of the “count” column tells where the problem is, the “class” column tells you with what. UNDO is rollback segments, DATA is data base buffers.

Question:: If you see contention for library caches how can you fix it?
Increase the size of the shared pool.

Question:: If you see statistics that deal with “undo” what are they really talking about?
Rollback segments and associated structures.

Question:: If a tablespace has a default pctincrease of zero what will this cause (in relationship to the smon process)?

The SMON process won't automatically coalesce its free space fragments.

Question:: If a tablespace shows excessive fragmentation what are some methods to defragment the tablespace? (7.1,7.2 and 7.3 only)

In Oracle 7.0 to 7.2 The use of the ‘alter session set events ‘immediate trace name coalesce level ts#’;’ command is the easiest way to defragment contiguous free space fragmentation. The ts# parameter corresponds to the ts# value found in the ts\$ SYS table. In version 7.3 the ‘alter tablespace coalesce;’ is best. If free space isn't contiguous then export, drop and import of the tablespace contents may be the only way to reclaim non-contiguous free space.

Question:: How can you tell if a tablespace has excessive fragmentation?

If a select against the dba_free_space table shows that the count of a tablespaces extents is greater than the count of its data files, then it is fragmented.

Question:: You see the following on a status report:

redo log space requests 23

redo log space wait time 0

Is this something to worry about? What if redo log space wait time is high? How can you fix this?

Since wait time is zero, no. If wait time was high it might indicate a need for more or larger redo logs.

Question:: What can cause a high value for recursive calls? How can this be fixed?

A high value for recursive calls is caused by improper cursor usage, excessive dynamic space management actions, and or excessive statement re-parses. You need to determine the cause and correct it By either relinking applications to hold cursors, use proper space management techniques (proper storage and sizing) or ensure repeat queries are placed in packages for proper reuse.

Question:: If you see a pin hit ratio of less than 0.8 in the estat library cache report is this a problem? If so, how do you fix it?

This indicates that the shared pool may be too small. Increase the shared pool size.

Question:: If you see the value for reloads is high in the estat library cache report is this a matter for concern?

Yes, you should strive for zero reloads if possible. If you see excessive reloads then increase the size of the shared pool.

Question:: You look at the dba_rollback_segs view and see that there is a large number of shrinks and they are of relatively small size, is this a problem? How can it be fixed if it is a problem?

A large number of small shrinks indicates a need to increase the size of the rollback segment extents. Ideally you should have no shrinks or a small number of large shrinks. To fix this just increase the size of the extents and adjust optimal accordingly.

Question:: You look at the dba_rollback_segs view and see that you have a large number of wraps is this a problem?

A large number of wraps indicates that your extent size for your rollback segments are probably too small. Increase the size of your extents to reduce the number of wraps. You can look at the average transaction size in the same view to get the information on transaction size.

Question:: In a system with an average of 40 concurrent users you get the following from a query on rollback extents:

ROLLBACK CUR EXTENTS

R01 11

R02 8

R03 12

R04 9

SYSTEM 4

You have room for each to grow by 20 more extents each. Is there a problem? Should you take any action?

No there is not a problem. You have 40 extents showing and an average of 40 concurrent users. Since there is plenty of room to grow no action is needed.

Question:: You see multiple extents in the temporary tablespace. Is this a problem?

As long as they are all the same size this isn't a problem. In fact, it can even improve performance since Oracle won't have to create a new extent when a user needs one.