



SQL PERFORMANCE TUNING IN ORACLE 10g AND 11g

Preeti Singh Bhadoria^{#1}, Ayushi Chhabra^{#2}, Sanjay Ojha^{#3}

School of Management,

Centre for Development of Advanced Computing (CDAC), Noida, Uttar Pradesh, INDIA

Abstract: Oracle Database 10g introduced SQL Advisor to simplify application and provides for SQL tuning which was considered as an advantage over the already existing Oracle 9i. Oracle 10g had greater advantages when it came to performing the root-cause analysis of one or SQL statements. It also involved one of the major disadvantages of involving several manual steps. This came in as major problem to the DBA's who had the tedious task of reviewing and testing the same, themselves. This issue was resolved with the introduction of the Oracle 11g that had the feature of the Automatic SQL Tuning, Real-time SQL Monitoring and Partition Advisor component of SQL Access Advisor that improved the performance problems without the DBA intervention.

Index Words: Automatic SQL Tuning, Automatic Workload Repository, Manual Tuning, SQL Tuning Advisor.

I. INTRODUCTION

Among the most challenging jobs that are often witnessed by the DBA's, the task of the SQL Tuning is the most challenging and the most sought after as far as the skill set of the DBA is concerned. To fulfill this task it is required that the DBA should possess an in depth knowledge of SQL and also should be able to recommend a solution for any query which might have been witnessed during the procedure.

A major leap in this regard was set forth with the introduction of the feature of SQL Tuning Advisor which was a feature introduced with Oracle version 10g. This advisor makes numerous recommendations on how to streamline everything from SGA configuration to individual SQL statements thereby reducing the burden of the DBA's. Though this was a huge improvement from the earlier versions like the Oracle 9i but still had a major drawback in terms of its manual handling of the processes that made the work of the DBA's tedious and cumbersome.

In order to overcome these issues the Oracle 11g was introduced with the new feature of Automatic SQL Tuning. Oracle automatically runs the SQL Tuning Advisor on selected high-load SQL statements from the Automatic Workload Repository (AWR) that qualifies as tuning candidates. This task, called Automatic SQL Tuning, runs in the default maintenance windows on a nightly basis. Automatic SQL tuning runs by default for at most one hour during a maintenance window. Automatic SQL Tuning needs one-time configuration for accepting automatically SQL profiles [1] on the high-load SQL.

II. MANUAL TUNING IN 10g vs. AUTOMATIC TUNING IN 11g

Performance Tuning is actually termed as the process of improving the system performance. The purpose for tuning is to reduce resource consumption or to reduce the elapsed time for an operation to complete. Either way, the goal is to improve the effective use of a particular resource.

The Figure1 given below shows clearly the entire working schedule of the SQL Tuning Advisor as it works in Oracle 10g [2].

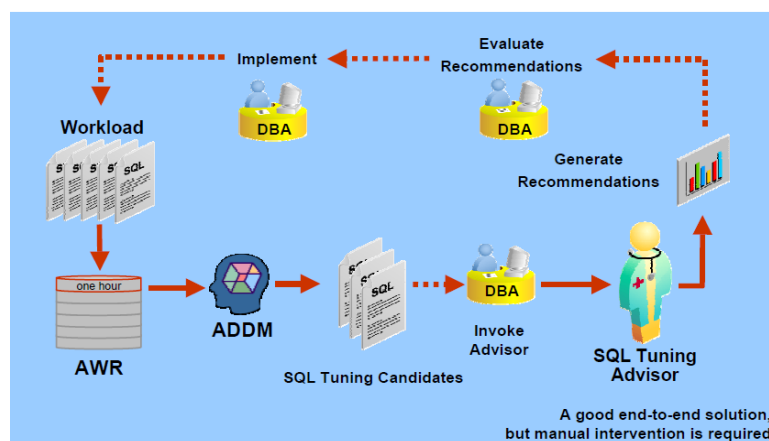


Figure 1: SQL Tuning in Oracle Database 10g

In 10g: The Figure 1[2] shows the Automatic workload Repository (AWR) that is responsible for collecting, processing, and maintaining performance statistics for the database. If we use services with our database, then AWR tracks metrics at the service level. The Automatic Database Diagnostic Monitor (ADDM) on the other hand is responsible for analyzing the AWR data on a regular basis, then locates the root causes of performance problems, provides recommendations for correcting any problems and identifies non-problem areas of the system. An ADDM analysis is performed every time an AWR snapshot is taken and the results are saved in the database.

However, it is important to realize that tuning is an iterative process, and fixing one problem can cause the bottleneck to shift to another part of the system. Even with the benefit of ADDM analysis, it can take multiple tuning cycles to reach acceptable system performance. ADDM benefits apply beyond production systems; on development and test systems, ADDM can provide an early warning of performance issues.

In 11g: Oracle 11g utilizes an automatic mechanism for performance tuning. This is shown in the Figure 2[3] as under:

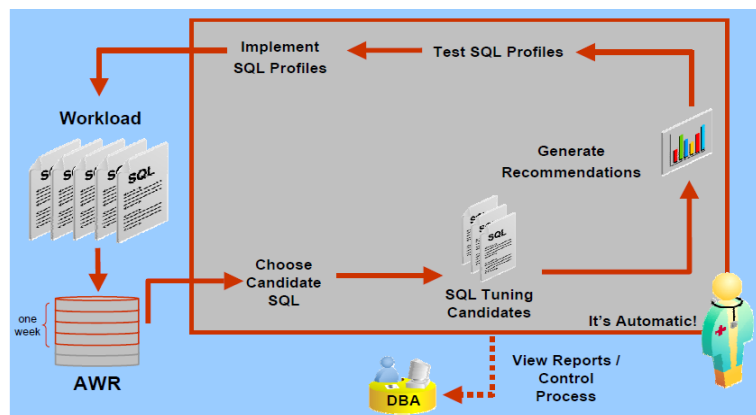


Figure 2: Automatic SQL Tuning in Oracle 11g

The Oracle 11g resolves the problem that is witnessed by the earlier versions of Oracle. This reduces the continuous intervention of the DBA that was seen in the earlier versions. Tuning in the case of 11g is generally of two main types [4]:

1. *Proactive Monitoring:*

Proactive monitoring usually occurs on a regularly scheduled interval, where several performance statistics are examined to identify whether the system behavior and resource usage has changed. Usually, monitoring does not result in configuration changes to the system, unless the monitoring exposes a serious problem that is developing.

2. *Bottleneck Elimination*

Tuning usually implies fixing a performance problem. However, tuning should be part of the life cycle of an application—through the analysis, design, coding, production, and maintenance stages. Often, the tuning phase is left until the database is in production. At this time, tuning becomes a reactive process, where the most important bottleneck is identified and fixed.

III. TOOLS UTILIZED IN SQL TUNING IN 10g AND 11g

The tools that are mainly responsible for improving the SQL Tuning in the two databases are [4] [5]:

a) *EXPLAIN PLAN:*

The statement displays execution plans chosen by the Oracle optimizer for SELECT, UPDATE, INSERT, and DELETE statements. A statement's execution plan is the sequence of operations Oracle performs to run the statement. The row source tree is the core of the execution plan.

The EXPLAIN PLAN results let you determine whether the optimizer selects a particular execution plan, such as, nested loops join. It also helps you to understand the optimizer decisions, such as why the optimizer chose a nested loop join instead of a hash join, and lets you understand the performance of a query.

b) *V\$SQL_PLAN View:*

V\$SQL_PLAN contains the execution plan for every statement stored in the cursor cache. Its definition is similar to the PLAN_TABLE.

The advantage of V\$SQL_PLAN over EXPLAIN PLAN is that you do not need to know the compilation environment that was used to execute a particular statement. The V\$SQL_PLAN_STATISTICS view provides the actual execution statistics for every operation in the plan, such as the number of output rows and elapsed time. All statistics, except the number of output rows, are cumulative. V\$SQL_PLAN_STATISTICS_ALL view

enables side-by-side comparisons of the estimates that the optimizer provides for the number of rows and elapsed time.

c) SQL TRACE and TKPROF:

The SQL Trace facility and TKPROF are two basic performance diagnostic tools that can help you monitor and tune applications running against the Oracle Server. The SQL Trace facility and TKPROF let you accurately assess the efficiency of the SQL statements an application runs. The SQL Trace facility provides performance information on individual SQL statements. We can enable the SQL Trace facility for a session or for an instance. When the SQL Trace facility is enabled, performance statistics for all SQL statements executed in a user session or in the instance are placed into trace files.

The additional overhead of running the SQL Trace facility against an application with performance problems is normally insignificant compared with the inherent overhead caused by the application's inefficiency.

d) STATSPACK:

STATSPACK is a set of performance monitoring and reporting utilities provided by Oracle starting from Oracle 8i and above. STATSPACK provides improved BSTAT/ESTAT functionality, though the old BSTAT/ESTAT scripts are still available. Although AWR and ADDM provide better statistics than STATSPACK users that are not licensed to use the Enterprise Manager Diagnostic Pack should continue to use STATSPACK.

e) Automatic Query Optimizer:

A special feature in 11g [4] is that when SQL statements are executed by the Oracle database, the query optimizer is used to generate the execution plans of the SQL statements. The query optimizer operates in two modes: a normal mode and a tuning mode.

1) **Normal Mode:** In normal mode, the optimizer compiles the SQL and generates an execution plan. The normal mode of the optimizer generates a reasonable execution plan for the vast majority of SQL statements. Under normal mode, the optimizer operates with very strict time constraints, usually a fraction of a second, during which it must find a good execution plan.

2) **Tuning Mode:** In tuning mode, the optimizer performs additional analysis to check whether the execution plan produced under normal mode can be improved further. The output of the query optimizer is not an execution plan, but a series of actions, along with their rationale and expected benefit for producing a significantly superior plan. When running in the tuning mode, the optimizer is referred to as the 'Automatic Tuning Optimizer'. The Automatic Tuning Optimizer performs four types of tuning analysis [4]:

- i. Statistics Analysis
- ii. SQL Profiling
- iii. Access Path Analysis
- iv. SQL Structure Analysis

i) Statistics Analysis: The query optimizer relies on object statistics to generate execution plans. If these statistics are stale or missing, the optimizer does not have the necessary information it needs and can generate poor execution plans. The Automatic Tuning Optimizer checks each query object for missing or stale statistics, and produces two types of output:

- Recommendations to gather relevant statistics for objects with stale or no statistics. Because optimizer statistics are automatically collected and refreshed, this problem may be encountered only when automatic optimizer statistics collection has been turned off.
- Auxiliary information in the form of statistics for objects with no statistics, and statistic adjustment factor for objects with stale statistics.

This auxiliary information is stored in an object called a SQL Profile.

ii) SQL Profiling: The query optimizer can sometimes produce inaccurate estimates about an attribute of a statement due to lack of information, leading to poor execution plans. Traditionally, users have corrected this problem by manually adding hints to the application code to guide the optimizer into making correct decisions. The Automatic Tuning Optimizer creates a profile of the SQL statement called a SQL Profile, consisting of auxiliary statistics specific to that statement. The query optimizer under normal mode makes estimates about cardinality, selectivity, and cost that can sometimes be off by a significant amount resulting in poor execution plans. The output of this type of analysis is a recommendation to accept the SQL Profile. A SQL Profile, once accepted, is stored persistently in the data dictionary. Note that the SQL Profile is specific to a particular query. If accepted, the optimizer under normal mode uses the information in the SQL Profile in conjunction with regular database statistics when generating an execution plan.

iii) Access Path Analysis: Indexes can tremendously enhance performance of a SQL statement by reducing the need for full table scans on large tables. Effective indexing is a common tuning technique. The Automatic Tuning Optimizer also explores whether a new index can significantly enhance the performance of a query. If such an index is identified, it recommends its creation.

Because the Automatic Tuning Optimizer does not analyze how its index recommendation can affect the entire SQL workload, it also recommends running the SQL Access Advisor utility on the SQL statement along with a

representative SQL workload. The SQL Access Advisor looks at the impact of creating an index on the entire SQL workload before making any recommendations.

iv) SQL Structure Analysis: The Automatic Tuning Optimizer identifies common problems with structure of SQL statements that can lead to poor performance. These could be syntactic, semantic, or design problems with the statement. In each of these cases the Automatic Tuning Optimizer makes relevant suggestions to restructure the SQL statements. The alternative suggested is similar, but not equivalent, to the original statement. For example, the optimizer may suggest to replace UNION operator with UNIONALL or to replace NOTIN with NOTEXISTS. An application developer can then determine if the advice is applicable to their situation or not. For instance, if the schema design is such that there is no possibility of producing duplicates, then the UNIONALL operator is much more efficient than the UNION operator. These changes require a good understanding of the data properties and should be implemented only after careful consideration.

f) SQL Tuning Advisor: The SQL Tuning Advisor takes one or more SQL statements as an input and invokes the Automatic Tuning Optimizer to perform SQL tuning on the statements. The output of the SQL Tuning Advisor is in the form of an advice or recommendations, along with a rationale for each recommendation and its expected benefit. The recommendation relates to collection of statistics on objects, creation of new indexes, restructuring of the SQL statement, or creation of a SQL profile. You can choose to accept the recommendation to complete the tuning of the SQL statements.

Oracle Database automatically runs the SQL Tuning Advisor on selected high-load SQL statements from the Automatic Workload Repository (AWR) that qualifies as tuning candidates. This task, called Automatic SQL Tuning, runs in the default maintenance windows on a nightly basis. Attributes can be customized for the maintenance of windows, including start and end time, frequency, and days of the week.

IV. CONCLUSION

To help simplify SQL performance tuning, Oracle introduced the SQL Tuning Advisor in 10g. It made many performance issues much easier to fix, but was lacking an automation framework: users still had to run the advisor manually when performance problems happened and accept the advisor's recommendations. However, with the release of the Oracle 11g was introduced Automatic Query Optimizer. Query optimization is the refining process in database administration and it helps to bring down speed of execution. This tool attempts to determine the most efficient way to execute the given query by considering the possible query plans. This tool hence provides an edge over the other tools as it leads to faster processing of the query, helps in reduction of the cost per query, results in high performance of the system, reduces stress on the database thus leading to efficient usage of the database engine due to which lesser memory is consumed. A fully self-tuning database capable of fixing or preventing every performance problem is a long way away to be sure. Thus, it can be said that the automatic query optimizer leads to better performance factors which were missing in the earlier versions of Oracle databases.

ACKNOWLEDGEMENTS

It is a well known fact that no task is achieved single handedly. It is always a collective task that is rewarded. Hence, in the present task too there are a number of people who are responsible in bringing out this paper in its current form. We would first of all like to extend our gratitude the entire management at CDAC Noida for supporting and providing us with the opportunity to utilize such a platform. We, would also like to thank Ms. Mary Jacintha (HOD, School of Management) and all the respective faculty members for their constant support and guidance.

REFERENCES

- [1] Julian Dontcheff, "The Power of 11g Automatic SQL Tuning", <http://www.oracle.com/technetwork/database/manageability/automatic-sql-tuning-nokia-ow09-130740.pdf> ; Nov 3, 2013
- [2] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zait, M. Ziauddin, "Automatic SQL Tuning in Oracle 10g," in VLDB pp 1098-1109, 2004.
- [3] Peter Belknap, Sergey Koltakov, Jack Raitto, "DBA's New Best Friend: Advanced SQL Tuning Features of Oracle Databases 11g", <http://www.oracle.com/technetwork/database/manageability/sqltune-presentation-ow07-130395.pdf> ; Nov 3, 2013.
- [4] Oracle Corporation, "Oracle Database Performance Tuning Guide 11g Release 2(11.2)", September 2013.
- [5] Oracle Corporation, "Oracle Database Performance Tuning Guide 10g Release 2(10.2)", March 2008