Query Optimization In Data Warehouse

BY

P. Arpitha, Assoc Professor (MCA)

ABSTRACT:

A data warehouse is a collection of integrated databases designed to support a DSS. It is a collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is non-volatile and relevant to moment in time. Query optimization is the bottleneck of database application performance especially those which store history i.e. data warehouse. SQL is used as query language because most data warehouses are based on relational or extended relational database system. In order to reduce the complexity of the query generation process and in order to preserve portability to other database systems proposed semantic query optimization architecture is very useful. e. Query optimization is of great importance for the performance of a relational database, especially for the execution of complex SQL statements. A query optimizer determines the best strategy for performing each query.

1. Introduction

Query optimization is the bottleneck of database application performance especially those which store history i.e. data warehouse. Numerous research has been introduced in the area of optimizing query performance, however a lot of research focused on online transaction processing (OLTP) database applications rather than data warehouse applications. SQL is used as query language because most data warehouses are based on relational or extended relational database system. As the information requests of the users are likely to be very complex. In order to reduce the complexity of the query generation process and in order to preserve portability to other database systems proposed semantic query optimization architecture is very useful. First, we offer some guiding principles for query optimization:

2.1. Understand how your database is executing your query

Nowadays all databases have their own query optimizer, and offers a way for users to understand how a query is executed. For example, which index from which table is being used to execute the query? The first step to query optimization is understanding what the database is doing. Different databases have different commands for this. For example, in MySQL, one can use "EXPLAIN [SQL Query]" keyword to see the query plan. In Oracle, one can use "EXPLAIN PLAN FOR [SQL Query]" to see the query plan.

2.2. Retrieve as little data as possible:

The more data returned from the query, the more resources the database needs to expand to process and store these data. So for example, if you only need to retrieve one column from a table, do not use 'SELECT *'.

2.3. Store intermediate results:

Sometimes logic for a query can be quite complex. Often, it is possible to achieve the desired result through the use of sub queries, inline views, and UNION-type statements. For those cases, the intermediate results are not stored in the database, but are immediately used within the query. This can lead to performance issues, especially when the intermediate results have a large number of rows.

The way to increase query performance in those cases is to store the intermediate results in a temporary table, and break up the initial SQL statement into several SQL statements. In many cases, you can even build an index on the temporary table to speed up the query performance even more. Granted, this adds a little complexity in query management (i.e., the need to manage temporary tables), but the speedup in query performance is often worth the trouble. Below are several specific query optimization strategies.

- **Use Index**
  Using an index is the first strategy one should use to speed up a query. In fact, this strategy is so important that index optimization is also discussed.
- **Aggregate Table**
  Pre-populating tables at higher levels so less amount of data need to be parsed.
- **Vertical Partitioning**
  Partition the table by columns. This strategy decreases the amount of data a SQL query needs to process.
- **Horizontal Partitioning**
  Partition the table by data value, most often time. This strategy decreases the amount of data a SQL query needs to process.
- **De normalization**
  The process of de normalization combines multiple tables into a single table. This speeds up query performance because fewer table joins are needed.
- **Server Tuning**
  Each server has its own parameters, and often tuning server parameters so that it can fully take advantage of the hardware resources can significantly speed up query performance.

2. Need and importance of query optimization:

Query optimization is of great importance for the performance of a relational database, especially for the execution of complex SQL statements. A query optimizer determines the best strategy for performing each query. The query optimizer chooses, for example, whether or not to use indexes for a given query, and which join techniques to use when joining multiple tables. These decisions have a tremendous effect on SQL performance, and query optimization is a key technology for every application, from operational Systems to data warehouse and analytical systems to content-management systems. In present scenario data warehouses & mining turned out to be the common basis for the integration and analysis of data in modern enterprises. Data mining based applications are used to analyze data on the operational level as well as on the strategic level. This includes techniques like online analytical processing (OLAP) and data mining. Additional tools are used for the preprocessing and integration of data from different sources. A lot of work has been done on data warehousing & data mining and their optimization. Query processing and optimization process work together to execute any kind of queries. Query processing is concerned with execution of query or refers to the activities involved in extracting data from a data warehouse. Query processing determines what data is to be fetched but does not define the method by which the data manager searches the database. On the other hand, optimization process deals with the efficiency of

3. Objectives:

The objectives and the purpose of query optimization is faster response to queries. The semantic optimizer knows more about its data rather than user. Therefore it can replace the user's query with a different query which will generate the same result set efficiently in less time. The new query is faster because it will do less work when extracting the selected result tuples from the data base.. we focus on a model of Optimizer Architecture for data Warehousing based (decision support systems) application. According to this model, the application generates a sequence of SQL statements, which is processed by the (OLAP Server) DWDBS.

4. Methodology:

SQL is used as query language because most data warehouses are based on relational or extended relational database system. As the information requests of the users are likely to be very complex. In order to reduce the_complexity of the query generation process and in order_to preserve portability to other database systems proposed semantic query optimization architecture is very useful.

5. Conclusion:

Optimization is much more than transformations and query equivalence. .Despite many years of work, significant open problems remain. However, it is necessary for making effective contribution to the area of query optimization. understanding of the existing engineering framework.

6.Reference

[1] Reema Theraja. Data Warehousing. Oxford University Press, 2009.

[2] Arun K Pujari, Data mining Techniques University press 2nd Edn. 2009.