

An Overview of Query Optimization Techniques in Database Systems

Pooja Wankhade¹, Dr. Vaishali Deshmukh²

¹Dept. Of Computer Sci. & Engg., PRMITR, Badnera, Maharashtra, India

²Prof. Dept. of Computer Sci. & Engg., PRMITR, Badnera, Maharashtra, India

Abstract:- Nowadays databases can store tremendous amount of information. At the beginning of database creation, the scale of data related to applications is small, the load on the database is light, so the performance of database system may seem satisfied but followed by the increase in data, the process consumes more resources and the speed of database system becomes slower. The amount of data keeps growing every second, queries keep on getting more and more complex and database operations are still expected to work quickly. The database encounters performance degradation. At this stage, database performance becomes a major concern. This initiates a need of query optimization.

This paper discusses about the importance, objectives and different approaches to query optimization. In this paper, we try to summarize some of the helpful query optimization techniques focused on common query constructs.

Key Words: Query optimization, Relational databases, Database tuning, Group by, Joins, Nested sub-query, Indexes

1. INTRODUCTION

Due to the huge growing applications, tremendous amount of data is generated every day. The size of the databases is increasing exponentially. Despite of the increasing data size, we expect that the database queries return answer in minimum time. As the queries become complex, it becomes necessary to optimize them to arrive at the results quickly. Otherwise, we would be spending most of our time to wait for the results only. Here query optimization proves to be of great help.

The simplest definition of query optimization is adopting different techniques for reducing running time of database queries. Its objective is to increase the speed and efficiency with which data is fetched/added from/to the database. Query optimization aims to allow us get the accurate result in the smallest amount of time. Optimization has always been and still is a very important topic in database systems.

The goal of query optimization is to minimize the response time of queries by making the best use of our system resources. It is very important for large scale database applications for efficient use of their system resources.

This paper is organized as follows. Section 1 discusses about the background and objectives of query optimization. Section 2 of literature survey summarizes the work related

to query optimization techniques. Section 3 lists some of the optimization techniques in brief. Finally section 4 concludes the paper.

1.1 Background

Query optimization is an important area of research in relational database management systems. In simple terms, a query is a request for information from a database. It can be as simple as "return list of all cities in a country" or somewhat complex as "return the average age of all residents having income greater than 20LPA and residing in metro cities of a country". The result for the query is obtained by manipulating the database such that it yields the requested information. This result can be obtained from a database by accessing it in various methods, using multiple data-structures, and in different sequences. Each method may require different processing time. The processing time may vary largely depending on the way used.

A query plan is a sequence of steps used to manipulate data in a database management system. There may be different query plans for the same query. One query plan may involve scanning the whole table to get the desired data while another may simple use index to directly fetch the requested record. Different query plans may give the same correct result but may require different processing time.

1.2 Objectives

The objectives of query optimization are –

- Achieve maximum throughput in minimum response time.
- Utilize minimum possible system resources to get desired result of a query
- Allow the system to process more queries in the same amount of time reducing processing time by optimizing badly written queries
- Reduce the amount of hardware wear and tear and improve efficiency by consuming less memory and power.

2. LITERATURE SURVEY

Query optimization is a wide research area in database field. A lot of research has been done and so much work is still

going on related to this topic. In following section, we try to summarize some of optimization techniques focused on group-by, joins and nested sub-queries, Having clause, indexes, etc.

Abhijit Banubakode, Virandra Dakhode, "Comparative Analysis of Query Optimization in the Object-Oriented Database & Relational Databases Using Clauses[1]" presented comparative analysis of query optimization for relational databases and object oriented database based on cost, cardinality and no of bytes. They focus on queries using where, group-by and having clauses. In this paper, they review the optimization technique, establishes the experimental setup and explains the query evaluation aspects. Their paper concludes that first if the group-by clause applied before conditional statement then there is significant cost reduction. The logic behind this optimization is based on the fact that a group-by reduces the cardinality of a relation, therefore if group by is evaluated early, it could reduce the cost of subsequent joins.

They explained this using an example query that computes count of bank branches located in a particular city and total count of branches in each city. In normal approach where group-by is applied after condition i.e. first check the condition and then group on branch city, search time is more and CPU cost is high. The optimization technique is to group on branch city first and then check the condition. This way if group-by is applied before condition, search time and CPU cost are proven to reduce significantly.

Weipeng P. Yan, Per Ake Larson, "Performing Group by before join[2]" proposed a new strategy for processing SQL queries containing group-by namely pushing the group-by operation past one or more joins. They consider sql queries containing joins and group by. They say that even though the standard way of evaluating this type of queries is to perform all the join first and then the group by operation, it may be possible to perform group by early i.e. to push group by past one or more joins.

According to them, early grouping may reduce the query processing cost by reducing the amount of data participating in joins. They stated necessary and sufficient conditions for using their proposed transformation and also stated an algorithm for deciding whether a group by can pushed past a join. They also noted some tradeoffs of this transformation and pointed out that reverse of the transformation is also possible and can be beneficial sometimes. They showed that their strategy may result in significant savings in query processing time. They assumed that queries were not containing 'Having' clause.

Won Kim, "On Optimizing an SQL-like Nested Query[3]" suggest to transform the nested queries to equivalent, non-nested queries that existing optimizers are designed to process more efficiently. He develops algorithms which transform nested queries to equivalent, non nested queries.

The algorithms are based on alternative ways of interpreting the operations of queries which involve the four types of nesting, and may often improve the performance of nested queries by orders of magnitude.

He analyzes the fundamental structure of an SQL-like, block-structured data language and how implementation suffers from the use of the inefficient nested-iteration method. He shows that the algorithm improves the performance of nested queries drastically. He also combines these algorithms into a coherent strategy for completely processing a general query of arbitrary complexity.

Fazal Mithani, Sahista Machchhar and Fernaz Jasdanwala, "A novel approach for SQL query optimization[4]" proposed a model to convert the input user SQL query into an optimized SQL query and aims to give assurance of reduced query execution time. They propose a scheme/approach to take an input query and provide a detailed plan called a Query Execution Plan (QEP) which includes related missing indexes, Join ordering, multitable joins and performance issues with LIKE operator.

Their process of query optimization for datasets is described as the following. First of all input query need to be run & full scan of dataset has to be performed. Then get the information about database schema, number of joins, missing indexes, redundant usage of tables, etc. From this information, find the alternate way to replace the method of query processing as per proposed scheme which identifies the syntax, constraint information, join aggregation and freeze some of data. Then by using all these rewrite the input query. Finally run the query and compare it with the input query. It gives surety about performance by an optimized execution. This proposed schema represents how the simple query is going to be converted in to optimized query in very accurate proper way with minimum amount of time and it will reduce the cost of execution and increase the speed of retrieval of data.

D.Saisanguansat and P.Jeatrakul," Optimization Techniques for PL/SQL [5]" presented optimization yechniques for PL/SQL like using index, out number technique, reducing sub-queries, etc. Using index technique is similar to the use of indexes in a book. Outnumber technique is to reuse the data values instead of repetitive calculation. Reducing sub-queries is to reduce the number of sub-queries to minimum possible count. They also presented combination technique which involves combining multiple optimization techniques.

They presented ten such optimization techniques and showed that each technique can improve the performance. Their results show that these techniques increase the efficiency by at least 9.88% to 732.25%. Using index techniques was proven to be the best effective one.

Jean Habimana, "Query Optimization Techniques - Tips For Writing Efficient And Faster SQL Queries[6]" proposed various general techniques that we can use to try to optimize our database queries. This paper does not focus on, in-depth analysis of database but simple query tuning tips & tricks which can be applied to gain immediate performance gain.

He suggested various tips like Using Column Names Instead of * in a SELECT statement, avoiding including a HAVING clause in SELECT statements, Un-nesting sub queries, avoiding using OR in join conditions, use of UNION ALL in place of UNION, avoiding functions on the right hand side of the operator, removing any redundant mathematics, etc.

3. SOME QUERY OPTIMIZATION TECHNIQUES

- Predicate pushdown
 - Perform predicate filtering before join to reduce size of join
- Perform projections early to create smaller tuples and reduce intermediate results.
 - Keep only required join attributes
- Rewrite to un-nest sub queries and/or flatten queries
- Decompose nested query and store result in sub table
- Try to push group-by past one or more join operations
 - Reduce the query processing cost by reducing the amount of data participating in joins
- use LOAD DATA
 - Can be used to load table from text file, this is usually 20 times faster than using INSERT statements.
- Use single INSERT statement with as much possible multiple values rather than multiple single row INSERTS
 - Can be used for inserting many rows from the same client at the same time
- Try to use column names instead of * in SELECT statements
- Remove unnecessary tables from query joins
- Use WHERE to limit the size of result tables that are created with joins.
- Use LIMIT to sample Query Results
- Drop unused Indexes

4. CONCLUSIONS

Performing database operations in specific time frame is important for any application today. There exists multiple ways of writing a query to achieve an equivalent operation on dataset. However, there can be massive difference in performance in executing these operations depending on the method used. Queries scripted in non optimal ways can be improved to perform better. A query written in optimal way gives the results in the desired time and makes the application run faster.

Query optimization is a research area where there is lot of scope of development. It's necessary for any organization to utilize the resources and be at its peak performance. Query optimization helps to achieve these goals.

This paper summarizes the problem, need and importance of query optimization. In this paper, we discussed about objectives and approach to query optimization. This paper gives an overview of some of the helpful query optimization techniques related to common query constructs like group by, joins, nested sub-queries, etc.

REFERENCES

- [1] Abhijit Banubakode, Virandra Dakhode, "Comparative Analysis of Query Optimization in the Object-Oriented Database & Relational Databases Using Clauses, International Journal of Advanced Research in Computer Science and Software Engineering Dec.2014.
- [2] Weipeng P. Yan, Per Ake Larson, "Performing Group by before join", Proceedings of 1994 IEEE 10th International Conference on Data Engineering.
- [3] Won Kim, "On Optimizing an SQL-like Nested Query", ACM Transactions on Database Systems, Vol. 7, No. 3, September 1982.
- [4] Fazal Mithani, Sahista Machchhar and Fernaz Jasdanwala, "A novel approach for SQL query optimization", 2016 IEEE International Conference on Computational Intelligence and Computing Research.
- [5] D.Saisanguansat, P.Jeatrakul, "Optimization Techniques for PL/SQL", 14th International Conference on ICT and Knowledge Engineering (ICT&KE) 2016.
- [6] Jean Habimana, "Query Optimization Techniques - Tips For Writing Efficient And Faster SQL Queries", International Journal Of Scientific & Technology Research Volume 4, Issue 10, October 2015 ISSN 2277-8616 22.
- [7] Vamsi Krishna Myalapalli, ASN Chakravarthy and KeesariPrathap Reddy "Accelerating SQL Queries by Unraveling Performance Bottlenecks in DBMS Engine", 2015

International Conference on Energy Systems and Applications (ICESA 2015).

[8] Vamsi Krishna Myalapalli and Pradeep Raj Savarapu, "High Performance SQL", 11th IEEE International Conference on Emerging Trends in Innovation and Technology, December 2014, Pune, India.

[9] Matthias Jarke, Jurgen Koch, "Query Optimization in Database Systems", ACM Computing Surveys (CSUR) Volume 16 Issue 2, June 1984.

[10] Surajit Chaudhari, "An Overview of Query Optimization in Relational Systems", PODS '98 Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.

[11] Vamsi Krishna Myalapalli and Muddu Butchi Shiva, "An Appraisal to Optimize SQL Queries", IEEE International Conference on Pervasive Computing, January 2015, Pune, India.

[12] Vamsi Krishna Myalapalli, Bhupati Lohith Ravi Teja, "High Performance PL/SQL Programming", IEEE International Conference on Pervasive Computing, pp. 1 – 6, 8 -10 Jan.2015.

[13] K. Kiran Kumar, T.M. Santhi Sri and Voruganti Vamshi priya, "Introduction to Techniques of Query Processing and Optimization", IJIRAE, vol. no. 2, pp.40-45 march 2015.

[14] P.Gulutzan and T.Pelzer, "SQL Performance Tuning", (Addison-Wesley,2003. 2003)

[15] S.Feuerstein, B.Pribyl, "Oracle PL/SQL Programming" (O'Reilly Media, 2005. 2005)

[16]Optimizing INSERT Statements
<https://dev.mysql.com/doc/refman/8.0/en/insert-optimization.html>