International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Review Article

Query Optimization in Distributed Database: A Review

Yasmeen R. M. Umar[Å*] and Amit R. Welekar[Å]

[Å]Computer Science & Engineering, Rashtrasant Tukadoji Maharaj Nagpur University, India

Accepted 10 Nov 2014, Available online 01 Dec2014, **Vol.4, No.6 (Dec 2014)**

Abstract

Distributed database is emerging as a boon for large organizations as it provides better flexibility and ease compared to centralized database. As the data is growing over the distributed environment day by day, a better distributed management system is required to manage this large data. Query optimization is a process of finding out better query execution plan from multiple available options. As there a multiple sites in distributed database having parts of the data, query optimization is one of the challenging tasks in distributed database. In this review paper query optimization challenges in distributed database and its basic steps have been studied. And a review of some proposed systems has been done.

Keywords: *Distributed database, query optimization, query execution engine, semijoin, ant colony algorithm etc.*

1. Introduction

Database is a collection of files or tables (relations). These collection of data needs to be managed which is done by a system called Database Management System (DBMS). There are two ways two manage these data. Centralized database holds all data on a central computer, the database presents physically at one location. In centralized database approach the data is placed on central repository hence it is easy to access or extract data from multiple tables. The database query can be easily transformed into set of relational algebra's operation.

A distributed database is a database in which portions of the database are stored on multiple computers within a network. Though the data is distributed, database is still centrally administered as a corporate resource while providing local flexibility and customization. The network should allow the users to share the data; thus a user (or program) at location A must be able to access (and perhaps update) data at location B. The sites of a distributed system may be spread over a large area (e.g., a city or a country) or over a small area (e.g., a building or campus).

A major objective of distributed databases is to provide ease of access to data for users at many different locations. To achieve this, the distributed database system must provide location transparency, which means that a user (or user program) using data for querying or updating need not know the location of the data. As data is distributed over several sites it requires more efforts to transform a query and in distributed database.

2. Query Optimization

With distributed databases, as the data is distributed over different sites, the response to a query may require the DBMS to assemble data from several different sites (although with location transparency, the user is unaware of this need). A major task for the distributed database is how to process a query, which is affected by both the way a user, provides a query and the intelligence of the distributed DBMS to develop a sensible plan for processing.

2.1 Query Optimization Challenges

As the data is distributed at different sites it is more challenging to compute efficient query plan in distributed environment. These challenges can be summarized as follows.

- First step is to break a query in a distributed database environment into components that are isolated at different sites.
- Determine which site has the potential to yield the fewest qualified records as it affects the communication cost of the network. The least the data to transfer across the network, the less communication will be required.
- Then transfer this result to another site where additional work is performed.
- If there are more than two sites, they require even more complex analyses and more complicated heuristics to guide query processing.
- Effective cost model at each site to compute the cost.

The basic steps followed by a distributed DBMS to develop a query processing plan:

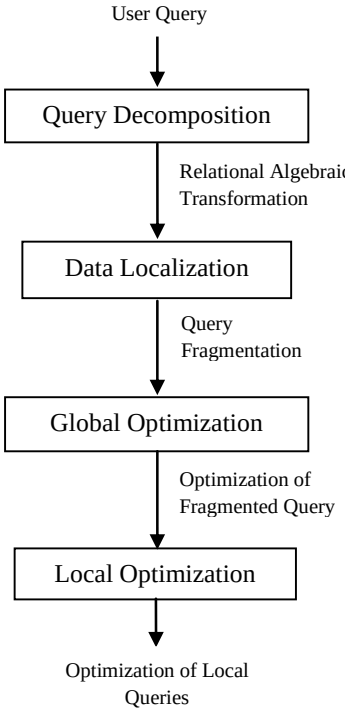
1. *Query decomposition:* In query decomposition, query is decomposed into simplified, structured, relational algebra form.

*Corresponding author: **Yasmeen R. M. Umar**

2. *Data localization:* Here, the query is transformed from a query referencing data across the network as if the database were in one location into one or more fragments that each explicitly reference data at only one site.

3. *Global optimization:* In this step, decisions are made about the order in which to execute query fragments, which site is efficient to move the data, and where parts of the query will be executed.

4. *Local Optimization:* When the fragmented query is sent to a particular site then that query will be executed and optimized locally.



The main components to optimize the query are query optimizer and query engine. The query engine produces the output by taking the input and operates on physical operators like SORT, NESTED LOOP JOIN, MERGE JOIN and INDEX SCAN etc. These physical operators construct a tree called parsed tree, which depicts the flow of data from one operator to the others in the form of edges moving back and forth to the nodes. Query optimizer receives a parsed tree of the SQL query as an input from the execution engine and produces the best possible or close to optimal execution plan out of the possible execution plans for the given query based on the least resource consumption. For a given query, there are many logical algebraic representations and there are many choices of physical operators to implement these logical representations in addition to the variation of response time of each plan. Therefore, it is not an easy task for a query optimizer to generate an efficient query plan.

3. Review of Proposed Strategies

Chaudhuri discussed the basics of query optimization and its key components like search spaces, an accurate cost estimation technique. An efficient algorithm has been proposed to generate the best execution plan. There are many stages when a query is submitted to a database server, the proposed phases are.

- The first phase is called parse tree,
- The intermediate phase is called logical operation tree
- And the final representation is called the operator tree.

For a single query there are number of possible logical trees to implement these trees, many combinations of physical operators are possible. The operator tree having least resource consumption would be the best or an optimal plan for the given query. For selecting best plan, the statistical information and execution cost are gathered and analysed. Statistical information like number of rows, memory requirement, joins and pages used by the relations etc. Statistical information can also be based on the columns if the columns are indexed. In order to select an inexpensive plan, an enumeration algorithm has been proposed to build an optimizer that adapts changes in the search spaces dues to the addition of new transformation or new physical operator. Such optimizers are called extensible optimizers, *e.g.*, Starburst and Volcano/Cascade.

Fan and Xifeng discussed the elements of distributed database and its features as follows:

1) Multiple computer equipment, connected by computer network.
2) Equipments required for computer network, a set of software of network communications.
3) GDBMS, LDBMS and CM are the elements of distributed database system, and in addition to a global user interface connected by the GDBMS, also have the autonomy site user interface connected by the site DBMS, has a separate site directory/dictionary.
4) Distributed database manager can be divided into two, one for the global database manager, and the other for the local or autonomous site database managers, which are collectively known as the local database manager.
5) Distributed database system software document, this is a set of software documents matched with the software, as well as a variety of system instructions and documents.

Also discussed the query optimization stages in distributed database. They proposed an algorithm to improve the semi-connected sub query optimization to reduce network communication cost. The basic principle of this optimization strategy is to use semi-connection operation to only transmit the data involved in the connection in the network as far as possible. As this algorithm focuses on semi connected queries only, it is less efficient for select query.

Chen and Yu more focused on reduction of communication cost. Detail study of joins and semijoins has been explored. Features of beneficial semijoin have been discussed. To reduce communication cost the combination of semijoin and join can be used effectively. The combination of semijoin and join as reducers results in substantially larger reduction on data transmission required. But the combination of semijoin and join should have a proper order to reduce the amount of data transmission. An algorithm has been developed to get an efficient order of Joins and semijoins. Stages have been discussed to include a join operation as a reducer in query processing. These stages are as follows:

1) Determine the effect of join operation.
2) Find out the best combination of gainful semijoin and pure join by evaluating their attributes. The semi joins that becomes beneficial due to the use of join and semijoin reducers are termed as gainful semijoins.
3) Then include this join and semijoin combination as reducer.

But the combination of semijoin and join should have a proper order to reduce the amount of data transmission. An algorithm has been developed to get an efficient order of Joins and semijoins. This technique only focuses on Join and semijoin queries.

Kossmann and Stocker proposed an algorithm to optimize query in distributed database based on Iterative Dynamic Programming (IDP). Algorithms based on Dynamic Programming produces good optimization results but for complex queries it becomes difficult to apply dynamic programming. Greedy algorithm is much faster compared to dynamic programming but it produces worse plans. Plans generated by greedy algorithm are not reliable always. Thus the combination of dynamic programming and greedy algorithm will generate efficient query plans. The algorithm is based on iteratively applying dynamic programming and can be seen as a combination of dynamic programming and the greedy algorithm. IDP-algorithms produce the best plans of all known algorithms in situations in which dynamic programming are not viable because of its high complexity. Query tunning is required in this approach and memory requirement is not considered.

The system proposed by XUE Lin contains multiple modules. These modules provide the query optimization flow in distributed environment.

- *User module* analyzes the user query and provides the interface to the user where user can opt for the ways results can be displayed.
- *Syntax analysis module* here lexical analysis and syntax analysis are done on global query. Query sentence is converted into syntax tree, on which semantic analysis is done.
- *Query tree conversion module* obtains Select data structure from the syntax analysis module and converts it into query tree and rewrite the query tree into an optimized query tree.
- *Optimization Module* receives the optimized query tree from *query tree conversion module* and maps it with physical operators, creating number of physical operator trees. It then calculates the cost of each physical operator tree and finds out the physical operator tree with minimum cost.
- *Order Processing Module* distributes the query to the corresponding server and returns the server processing results to the user. A local data dictionary is maintained with sentence table that stores mostly used results to avoid the transmission of large dat. This affects on the query efficiency. But as the size of table increases, the CPU processing time will also increase and memory requirement also increases.

As the dynamic programming and greedy approach could not provide efficient query execution plan, the latest approach for query optimization has been proposed for relational database based on ant colony algorithm. An ant colony algorithm is an efficient algorithm to find out the shortest path. Like Travelling Sales man Problem, ant colony can also be applied to relational database system. In this approach each relation is considered as one of the cities and ant colony algorithm is applied among the multiple cities to find out the shortest path.

A better execution time has been achieved through this approach. The same idea of ant colony can also be applied to distributed database system to achieve a better execution plan.

Conclusion

In this paper challenges with distributed database environment has been discussed and its basic steps has been studied. A review of some proposed system has been done to analyze the query optimization in distributed database system and their demerits to overcome in future work.

References

C.T. Yu and C.C. Chang (Dec. 1984), Distributed Query Processing ACM Computational Surveys, vol. 16, no.4, pp. 399-433.

P.A. Bernstein, N.Goodman, E.Wong, C. Reeve and J.B. Rothnie (Dec. 1981), Query Processing in a System for Distributed Database(SDD-1), ACM Trans. Database Sys, Vol.6, no. 4, pp 602-625.

Doshi P. and Raisinghani V. (April 2011), Review of Dynamic Optimization Strategies in Distributed Database, Electronics Computer Technology (ICECT), 3rd International Conference.

Surajit Chaudhuri, An Overview of Query Optimization in Relational Systems, Microsoft Research.

Fan and Xifeng (2010), Distributed Database System Query Optimization Algorithm Research, IEEE2010.

MS Chen, PS Yu (August 2005), Interleaving Join Sequence with Semijoins in Distributed Query Processing, IEEE Transactions, Issue 5, Vol. 3.

Donald Kossmann, Konrad Stocker, Iterative Dynamic Programming: A New Class of Query Optimization Algorithms.

XUE Lin (2009), Query Optimization Strategies and Implementation Based on Distributed Database, IEEE

Adel Alinezhad Kolaei and Marzieh Ahmadzadeh (October 2013), The Optimization of Running Queries In Relational Databases Using Ant-colony Algorithm, IJDMS Vol.5, No.5.