Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Issues in Query Processing and Optimization

Vineet Mehan[1], Kaushik Adhikary[2], Amandeep Singh Bhatia[3]
[1]*CSE, MAIT, Maharaja Agrasen University, H.P.*
[2]*CSE, MAIT, Maharaja Agrasen University H.P.*
[3] *CSE, MAIT, Maharaja Agrasen University H.P.*

Abstract—The paper identifies the various issues in query processing and optimization while choosing the best database plan. It is unlike preceding query optimization techniques that uses only a single approach for identifying best query plan by extracting data from database. Our approach takes into account various phases of query processing and optimization, heuristic estimation techniques and cost function for identifying the best execution plan. A review report on various phases of query processing, goals of optimizer, various rules for heuristic optimization and cost components involved are presented in this paper.

Keywords- Query processing; Optimization; Heuristic estimation

I. INTRODUCTION

Users of the database have less knowledge about the working of the database. So this burden of choosing the best query should be put on the DBMS/RDBMS and not on the user. Here comes the role of Query Processing and Query Optimization [1]. There are 'n' numbers of ways to run a query. A question arises which way is the best? To choose the best way we must know the following:

- How a Query is processed by DBMS/RDBMS.
- What are the different ways or plans in which a query can be formed?
- Which plan is the best among all the other plans?

II. PHASES IN QUERY PROCESSING

DDL commands are not processed by the query optimizer. Only DML commands are processed by the query optimizer [2]. Steps involved in Query processing include:

- Search Query
- Parsing and Validating
- Optimization
- Code Generation
- Query Execution
- Search Results

Search query is any SQL query for which optimization is to be done. Parser is a tool that transforms a query to structure. It checks for the correct syntax (i.e. Table Name, Attribute Names, Data types etc.). It resolves names and references and converts the query into parse tree/query tee. To simplify the query translation process query is broken into blocks [3]. Each query block consists of a SELECT, FROM, WHERE block along with some blocks with AND, GROUP BY and HAVING

clauses. Parser may also check if user is authorized to execute the query or not. Parsed query is then sent to the next step for Query Optimization.

Output of the parser acts as an input the query optimizer. Goal of optimizer can be any one or all of the following:
Goal 1: Minimize Processing Time
Goal 2: Minimize Response Time
Goal 3: Minimize Memory Used
Goal 4: Minimize Network Time

Once the query optimizer has determined the execution plan (the specific ordering of access routines). The code generator writes out the actual access routines to be executed. The query code is interpreted and passed directly to the runtime database processor for execution. It is also possible to compile the access routines and store them for later execution [4]. Query has been scanned, parsed, optimized, and (possibly) compiled. The runtime database processor then executes the access routines against the database. The results are returned to the application that made the query in the first place. Any runtime errors are also returned. When the query is executed, results are obtained to be displayed to the user.

III. HEURISTIC OPTIMIZATION TECHNIQUES

Heuristic refers to experience-based techniques for problem solving, learning, and discovery. The solution obtained may or may not be optimal. Under heuristic optimization one should identify the techniques which make our queries optimized [5]. Common Heuristic based technique is Rule of Thumb. The term originated with carpenters who used width of their thumbs for measuring rather than measuring scales. The main reason for such a measurement was on the basis of experience. When you are an experienced carpenter you think that your measurement is right. Various rules identified include:

a) Carry out Selection as early as possible.
b) Projections are executed as early as possible.
c) Cascading Selections(S) and Projections (P): When S and P are on the same operand then operations may be carried out together. It saves the cost of scanning a table more than once.
d) Optimal Ordering of Joins: Ordering of joins should be such that the results are small rather than large.
e) Combining certain Projections and Selections instead of a Join.
f) If there is more than one projection on the same table the projections should be carried out simultaneously.
g) Sorting is deferred as much as possible.

IV. COST FUNCTIONS

Several Cost components include:

- Access cost to secondary storage (hard disk).
- Storage Cost for intermediate result sets.
- Computation costs: CPU, memory transfers, etc. for performing in-memory operations.
- Communications Costs to ship data around a network. E.g., in a distributed or client/server database.

Access cost to secondary storage involves access cost in terms of number of rows and access cost in terms of memory [6]. Since the execution takes place step by step. Intermediate results are stored in temporary files/tables. Cost of accessing tables and memory is also involved. This accounts for Storage Cost for intermediate result sets. Every storage access involves certain kind of computation cost. E.g. when a particular application is running say Microsoft power point then CPU is being utilized. Similarly to access database for some queries certain computation cost is involved. If the model that we are using is client server based then communication cost is also involved. Bandwidth is required to fetch a heavy database. E.g. If a website works on Oracle database then, there may be certain dates on which to due to heavy traffic the site hangs up or becomes slow. For discussion let us take Cost function for SELECT statement and Cost function for JOIN.

Cost function for select means cost function for Selection. Selection in relation algebra accounts for conditions. So the cost function for selection depends upon the conditions as shown in Table 1.

Table1. Conditions in cost function for selection.

S. No.	Conditions in "where" clause
1	Attribute A = value v
2	Attribute A > value v
3	Attribute between value v1 and v2
4	Attribute A IN (List of values)
5	Attribute A IN Subquery
6	Attribute A condition C1 OR condition C2
7	Attribute A condition C1 AND condition C2
8	Attribute A is NOT NULL

Cost function for Join include: Nested-loop Join; Nested Index Join; Sort Merge Join/ Sort Scan Join and Hash Join. In Nested Loop Join the optimizer chooses one of the tables as the outer table. The other table is called the inner table. For each row in the outer table, optimizer finds all rows in the inner table that satisfy the join condition. Database compiler can only perform a sort-merge join for an equijoin. To perform a sort-merge join, following steps are required: Sort each row source to be joined; Rows are sorted on the values of the columns used in the join condition; Compiler then merges the two sources.

V. CONCLUSION

Query processing and Optimization is much more than merely choosing the best query plan. Designing effective and correct plan involves considering a number of factors which vary from initial phases to the cost functions involved. Even though the work has been done in the area of query processing and optimization but still there are significant open problems that exist. Nevertheless, a perceptive of the accessible engineering framework is essential for making effectual role to the area of query optimization.

REFERENCES

[1] S. Rahimi, F. Haug, "Distributed Database Management Systems:A Practical Approach", pp. 111-181, 2010.

[2] S. Bottcher, D. Bokermann, R.Hartel, "Generalizing and Improving SQL/XML Query Evaluation ", Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), 2012, pp. 441 - 449

[3] P. Doshi, V.Raisinghani, "Review of dynamic query optimization strategies in distributed database", 3rd International Conference on Electronics Computer Technology (ICECT), vol. 6,pp. 145 – 149, 2011.

[4] T. V. V. Kumar, V. Singh, A. K. Verma, "Generating Distributed Query Processing Plans Using Genetic Algorithm ", International Conference on Data Storage and Data Engineering (DSDE), 2010, pp.173-177.

[5] L. Antova, T. Jansen, C. Koch, D. Olteanu, "Fast and Simple Relational Processing of Uncertain Data ", 24th International Conference on Data Engineering, 2008, pp. 983 - 992

[6] C. Binnig, D. Kossmann, E. Lo, "Reverse Query Processing ", 23rd International Conference on Data Engineering, 2007, pp. 506-515.

