ISSN 2347-4289

Different Sql Query Optimizer Techniques

Shani.S.Das, Rejimoan .R

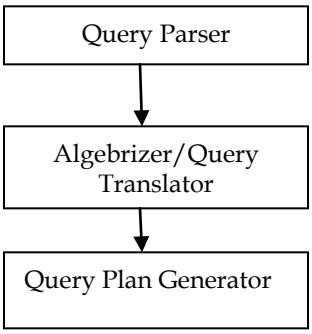
Mtech Student, Dept. of CSE, SCTCE, TVM, India ; Asst.Prof. Dept. of CSE, SCTCE, TVM, India
shani.sdas@gmail.com, rejimoan@gmail.com

ABSTRACT: Database contains huge amount of data. In order to make the query retrieval easier the content of the database is much organized. The query language used to retrieve the data stored in the database is SQL. In addition to the organized structure of data in the database there exists many query optimization techniques to make the query retrieval fast. SQL statements are declarative. That means SQL query can be written in different ways to retrieve the same query result. It is trivial to find which organization of the SQL query retrieve is query result with a minimum time. This paper discusses various optimizer approaches used for query retrieval from database and finds the better approach among them. The various query optimizer approaches discussed in this paper is Cost based ,Heuristic based and Learning based . Cost based approach focus on the statistics of the relations . Heuristic approach works based on certain predefined rules and Learning based approach works by correcting the incorrect statistics used to execute the query.

Keywords: SQL Query; Optimizers; Query Tuning

1 INTRODUCTION

The data stored in the database is retrieved through the SQL query .The query is submitted either by the developers or by the administrators. The query results are used by various applications by which most of them may be time critical. Hence the query execution time must be less so that the corresponding application. The query submitted by the user is passed to the query compiler. The functioning parts of a compiler includes ,



The query compiler consists of query parser, query translator and finally query optimizer. The function of the query parser is to check whether the query submitted by the user is syntactically correct or not. If the query submitted by the user is not syntactically correct then the query parser will return error.The parser output a parse tree if the query is syntactically correct. Then the query is passed to the query translator which translates the query to a relational algebraic expression. The expression so developed is passed to the query optimizer. The optimizer is a software which models how to execute the query based on various statistics. By using the statistics collected, the optimizer will generate an execution strategy what it thinks to be an optimized way to execute that particular query. The query is executed based on this execution strategy developed by the optimizer[1].This execution strategy which instructs the query compiler how to execute a query is termed as execution plan.Based on how the execution plans are generated, the query optimizers are classified as cost based and heuristic based optimizer[2][3]. This survey paper mainly focus on the different types of optimizers such as cost - based , heuristic - based and learning-based approaches .The cost based optimizers mainly focus on the execution plan with lowest cost by finding the cost of each individual operation in the submitted query. Heuristic based optimizers have certain rules and produce an execution strategy based on these rules. Learning optimizers will correct the incorrect statistics chosen by the previous execution plans[4][5]. There are optimizers which focus on the cost with heuristic approaches[6]. This paper contains four sections .Section II discusses different query optimization techniques. Section III make a comparison of these optimizer approaches and Finally section

2. QUERY OPTIMIZERTECHNIQUES

2.1 Cost – Based Approach

Cost - Based Query Optimizers, focus on the cost attributes related to the query execution. During the optimization is it needed to look at the data used for the query. But these cost-based query optimizers mainly focus on the syntax of the query not on the data. The cost components are [6][7] ,

- Access Cost - The cost for retrieving the data from the secondary storage devices
- Storage Cost - The cost related to the storage of intermediate results during the query execution
- Computation Cost - Cost related to the memory operations
- Communication Cost - Cost related to the shipping of data across the network

In general, if a query contains n number of operations then the operations can be performed in different orders which results in n! execution plans. The execution plans is a strategy which informs the compiler , by which order the operations is to be executed. The execution plan with the minimum cost is taken as the optimal plan for the query submitted by the user. The algorithm below discusses the cost – based approach for query optimization. Find all possible query plans for the query by changing the order of execution of different operations in the query. As a general procedure for optimizing do the following steps

- Perform all the simple selections first
- Perform the join next
- Perform the projection last

Algorithm 1 Cost –Based Approach

1. 1.Establish all the of the permissible plans (P1,P2,.....,Pn) of the query where each plan contains a set of operators O1 ,O2.....,Ok
2. Choose a plan out of these n plans
3. For each operation Oi mention all the access routines. These acess routines are the algorithms that are used to access and aggregate the data in the database
4. For each access routine of Oi ,estimate the cost. Select the access routine with the lowest cost
5. Repeat the step 2 untill an efficient access routine has been selected for each operation
6. Sum up all the cost of each access routine to determine a total cost for the plan
7. 7. Repeat step 2 through step 5 for each plan and chose a plan with lowest cost

For example, consider that the following operations exists in a query (one select (S1), two join (J1,J2) and one project(P1) operation)

- Enumerating Plans
  - Plan A - S1,J1,J2,P1
  - Plan B - S1,J2,J1,P1
- Select Plan A
- For each operation establish the access routines
  - S1 - Linear search and Binary Search
  - J1 - Nested loop join and Index join
  - J2 - Single loop join
- Choose the minimum cost access routines for each operation

S1 - Least cost access routine is binary search , 10 blocks

J1 - Least cost access routine is indexed join at a cost of 50 blocks

J2 - Least cost is single loop join , 20 blocks

Total cost = 10+50+20=80 blocks

- Plan B
  - S1- binary search ,5 blocks
  - J1 - Nested loop join ,20 blocks
  - J2 - Single loop join , 20 blocks

  Total cost = 5+20+20=45 blocks
- Hence the plan B has least cost

The main problem with this approach is that, the search space can become quite large depending on the complexity of the SQL query

2.2 Heuristic-Based Approach

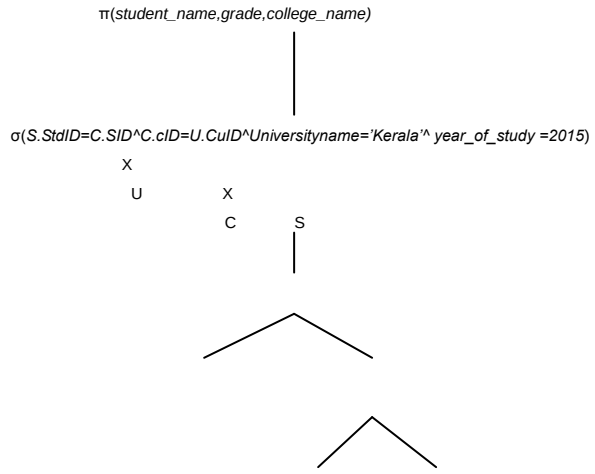
Heuristics optimizers works based on certain predefined rules. These optimizers try to minimize the access by reducing the number of tuples and number of columns to be accessed [7]. In this approach a query is represented as a tree like structure. Operations are at the interior nodes and data items are at the leaf nodes. There are a number of transformation rules that can be applied to transform a query. The algorithm2 discusses the heuristic based query optimization. It specifies general rules to generate an optimized execution plan for the submit ted query.

Algorithm 2 Heuristic – Based Approach

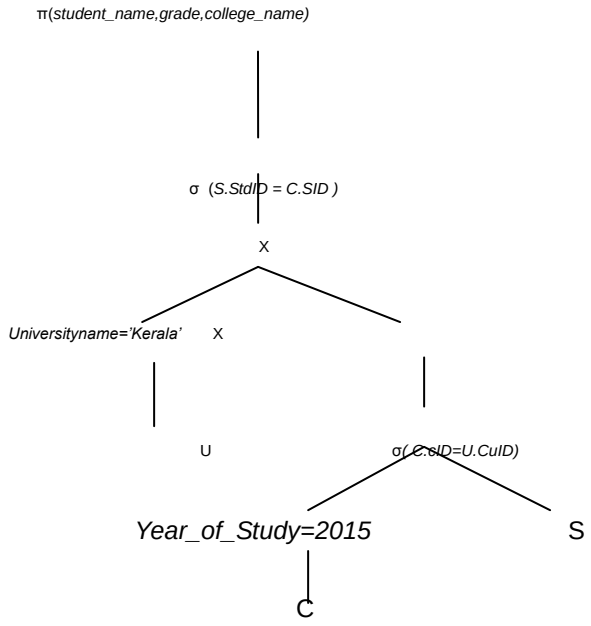
1. 1.Convert the query submitted by the user to an equivalent relational algebraic expression
2. Break up any select operations with conjunctive conditions into a cascade of select operations
3. As per the attributes involved in the selection condition move the select operation as down as possible
4. Perform the most restrictive condition in the leaf nodes by rearranging the leaf nodes
5. Combine a Cartesian product operation with a subsequent select operation in the tree into a join operation
6. 6. Move the project operation as far as down as possible by breaking the project condition

For example, Consider a Education schema which contains different universities, colleges, courses, students. Suppose aquery is given by the user to retrieve the names, grade of students belonging to colleges in Kerala university by the year 2015.

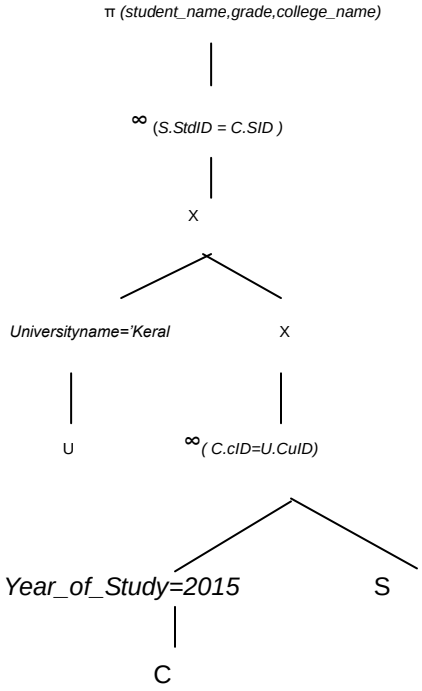
- Step 1 - An unoptimized query tree



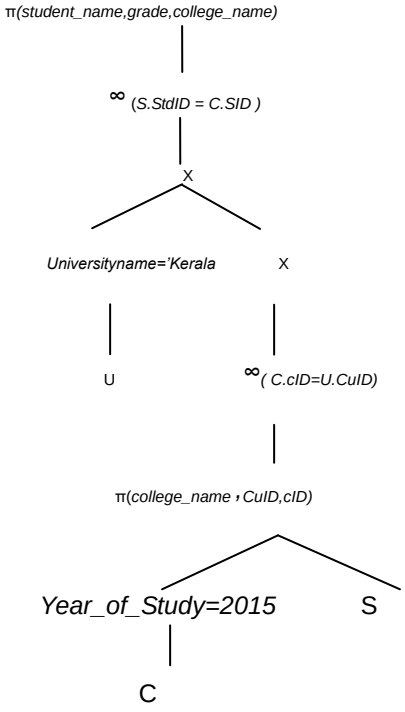
- Step 2 - Moving the restrictive operations to the leaf notes



- Step 3 - Combining the cartesian product and the selection operation



- Step 4 - Apply the project operation to the leaf nodes



• Finally an optimized execution plan is obtained

The main problem with the Heuristic – based optimizers is that the does not look into the statistics of the data. If there exists an index for the relational data, then this approach always use the index to execute the query but sometimes this will take more time.

2.3. Learning-Based Approach

The cost based and heuristic based approaches mainly focus on the existing statistics of the data in the database. There is a chance that the optimizer may chose the incorrect statistics for preparing the execution strategy. LEarning Optimizer[4][5] mainly focus on finding out whether optimizer has chosen incorrect statistics and if the optimizer has chosen an incorrect statistics the optimizer proposes another execution plan for the same query. This mainly focuses on learning from a modeling mistake at any point in the query execution strategy. It provides a different approach to correct the incorrect statistics. It monitors the optimizers actual cost with the estimated one on each step during the query execution .Compute the cost of changes and statistics that may be used for future reference.LEO introduces a feedback loop which contains the changes in the query plan and enhance the available information on a database for the query. This allows the optimizer to actually learn from its past mistakes. LEarning Optimizer consists of mainly four components to save the optimizers plan, a monitoring component, an analysis component, and a feedback exploitation component. The execution plans are captures at compile time, monitoring the costs estimates is done step by step during run-time and the feedback loop is integrated with the optimizer. Even though each component is independent to another there is a sequential flow within these components. The feedback loop works with as assumption that the future queries have the similar features to the previous queries[5]. The method of prediction can be used to improve the optimization of sql query much more[10][11].

3. Result And Analysis

Each of the approaches discussed earlier has its own advantages and disadvantages. The performance of these optimizers relies on the statistics used for the query execution as well as the execution plan generated. Consider a relational schema of a Company having relations Employee, Department, Project, Dept_locations. The test database used is Oracle. Suppose the query submitted by the user is

SELECT P.Pnumber,P.Dnum,E.Lname,E.Address,E.Bdate

FROM PROJECT as P ,DEPARTMENT as D,EMPLOYEE as E

WHERE P.Dnum=D.Dnumber and D.Dnumber=E.Dno;

The main goal is to find which approach is good among those described previously. Most of the database servers uses cost based approach for query optimization. There are optimizer which works on heuristics also. The time is used as criteria to analyse the performance of the query. If the time is less, the approach is best for that type of query. When the above query is submitted to a cost based optimizer it takes 8 ms to execute the query. When submitted to the heuristic based optimizer it took 9 ms to execute the same query.Different types of queries are submitted and the observations are as follows

Query Type	Query	Cost-Based	Heuristic-Based
Join Query	SELECT P.Pnumber, P.Dnum,E.Lname, E.Address,E.Bdate **FROM** PROJECT AS P ,DEPARTMENT as D,EMPLOYEE as E **WHERE** P.Dnum=D.Dnumber and D.Dnumber=E.Dno;	8 ms	9 ms
Correlated Nested Query	SELECT E.Fname, E.Lname FROM EMPLOYEE AS E WHERE EXISTS (SELECT * FROM DEPENDENTAS D WHERE E.Ssn=D.Essn ANDE.Sex=D.Sex AND E.Fname=D.Dependent_name);	12 ms	14 ms
Simple Query	SELECT ssn,Fname, Address FROM EMPLOYEE, DEPARTMENT WHERE Dname='HR' AND Dnumber=Dno;	2 ms	2 ms
Aggregate Query	SELECT SUM (Salary), MAX (Salary), MIN (Salary), AVG (Salary) FROM EMPLOYEE;	6ms	7 ms

From the above observation it can be summarized that for complex queries cost – based approach provides optimal query plan and take minimum time. The learning optimizer is an improved version of this which works by changing the incorrect statistics. So this is useful only when the optimizer uses incorrect statistics to work.

4. CONCLUSION

Among the functions of the DBMS the most critical one is to process a query in a timely manner. The importance of this aspect mainly relies on very large databases which contain trillions of records. The need of the faster and faster execution of the query increases day by day. Many works are progressing on this area relating to optimize the query submitted to the query engine. Query optimization is the most important task performed by the database administrator and designer to improve the overall performance of the query execution. Even though the cost based optimizer chooses an execution plan which is less expensive it does not look into the statistics of the data it is trying to optimize. The statistics is an important aspect in query optimization. Moreover the search space is large for the cost-based approach according to the complexity of the query submitted. The heuristic based approach relies on the certain predefined rules inadditon to the statistics. The main problem with this optimizer is that it often chooses the wrong index to execute the query plan. It consume a lot of time. The Learning optimizer figures out whether the optimizer has chosen wrong index and if so it corrects it by changing the statistics of relations and make adjustments in the execution plan

REFERENCES

[1] Grant Fritchey ,SQL Server Execution Plans, Simple Talk Publishing September 2012

[2] A. Hameurlain and F. Morvan Evolution of Query Optimization Methods Springer- Verlag Berlin Heidelberg pp. 211–242, 2009

[3] Michael L. Rupley, Jr. Introduction to Query Processing and Optimization,2005

[4] Michael Stillger,, Guy Lohman, Volker Markl, Mokhtar Kandil LEO – DB2's LEarning Optimizer Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[5] Dave Martin , Barry Thorn , Steve Tlusty , JaeHee Yang Preparing for Tuning the SQL Query Engine on DB2 for i5/OS Redbooks

[6] Saurabh Kumar,Gaurav Khandelwal,Arjun Varshney,Mukul Arora ,Cost- Based Query Optimization with Heuristics IJSCR Volume 2, Issue 9, September-2011

[7] Surajit Chaudhuri An Overview of Query Optimization in Relational Systems Microsoft Research

[8] An Oracle White Paper November 2010 SQL Plan Management in Oracle Database 11g

[9] Oracle 11g the complete reference

[10] Predicting Multiple Metrics for Queries: Better Decisions Enabled by Machine Learning, 2012

[11] Mert Akdere, Ugur Cetintemel, Matteo Riondato, Eli Upfal, Stanley B. ZdonikLearning-based Query Performance Modeling and Prediction 2012 IEEE 28th International Conference on Data Engineering