Optimization of Distributed queries based on cost complexities

Amandeep Kaur
IT Department, PCTE Group Of Institutes
#2173/2D, Street No. 7, Jammu Colony, Ludhiana [Punjab, India]
amandeep@pcte.edu.in

ABSTRACT: In DB area, may be centralized or decentralized, Query optimization is the hardest problem. A user can pass query in declarative way in form of SQL command or some other form, some execution strategy is required for Query performance tuning. Moreover a distributed database is a collection of independent cooperating centralized systems. Query processing in a distributed database requires transfer of data from one computer to another through a communication network. Query at a given site might require data from different (remote) sites. An optimizer can help in this regard as it can decide in which order a query should be executed. Cost is measured in terms of sum of I/O cost, CPU cost at each site and the cost of transferring data between sites. The complexity and cost increases with the increasing number of relations and amount of data transmission in the query. If number of relations and data transmission can be minimized, then cost can be automatically controlled. The key problem for query optimization in a distributed database is selection of the most cost effective plan to execute a query. Lots of research is going on in this area. Here this paper will focus on optimization model, application of optimization techniques on queries and cost analysis before and after applying Query optimization.

KEYWORDS: DB, Centralized, Decentralized, Query, SQL, Performance tuning, Distributed, Network, Remote, I/O, CPU, Site, Relation, Optimizer.

I. INTRODUCTION:

A database is physically distributed across distributed across different sites. Distribution is done using fragmentation namely Horizontal Fragmentation (Using Selection Operation), Vertical Fragmentation (Using Projection Operation), Mixed and Derived ones. Sometimes this leads to replication of data at different sites too. Fragmentation is useful as it allows placement of data in close proximity to its place of use. Amount of data handled by DBMS increases continuously from GBs to TBs and so on. Managing such a huge DB is not a critical task but most challenging is to develop an efficient Query processing technique, which involves retrieval of data from DB. There are many ways to execute a Query, but objective of Query Processing is to find cost effective technique. There comes the role of Query Optimizer. Actual objective of optimizer is to find a strategy close to optimal solution & to avoid bad strategies.

Cost can be minimized by amelioration or optimization where amelioration is the process of modifying a query with two expectations i.e. commutative of joins & unions and transformation of joins into semi join programs. On the other hand Query optimization is the process of selecting among alternative possibilities of the one, which can minimize the cost.

A. Important Aspects of Query Optimization:
   a. Distribution of access strategies among sites, which is based upon transmission only.
   b. Determination of access strategies at each site locally, same as centralized DB only.

B. Problems in Query Optimization:
   a. Materialization: Ensuring availability of Fragments or other schemata to execute Query and moreover minimizing access of global databases to avoid cost of data transmission over network.
   b. Order of execution: In which order various operations like joins, semi-joins, selection or projection can be executed to ensure optimization of Query. Sometimes amelioration applied on operator tree may not result better performance.
   c. Methods for executing Operations: Which alternative will be the best one to execute query in an optimal way. Whether to apply selection, projection, joins or semi-joins etc.

C. Solutions or Assumptions:
   a. Materialization is always taken as granted i.e. physical copies of fragments are available for query execution.
   b. It is assumed that order of execution is already optimized.
   c. Operations are clustered into local programs, otherwise it is system dependent so not considered or degraded.

D. Objectives of Query Optimization:
   a. To minimize CPU Processing, I/O cost and data transmission cost.
   b. To minimize amount of data transmission.
   c. To minimize distribution and local access strategy.
   d. To minimize the consumption of computing resources.
   e. To reduce the response time of a distributed query, which the time elapse between issuance and answer production of a query. Ultimately the main aim is to minimize the total time required to execute a Query.

The important objective is the cost factor, where transmission requirements are generally neutral. They typically employ a function for amount of data being transmitted over sites, but same is not true for CPU and I/O usage.

II. OPTIMIZATION MODELS:

A. Optimization Model in Centralized DB:

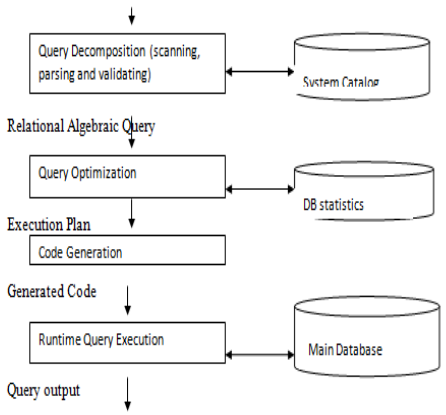

Fig. 1 Optimization model in centralized system

B. Optimization Model in Distributed Database:

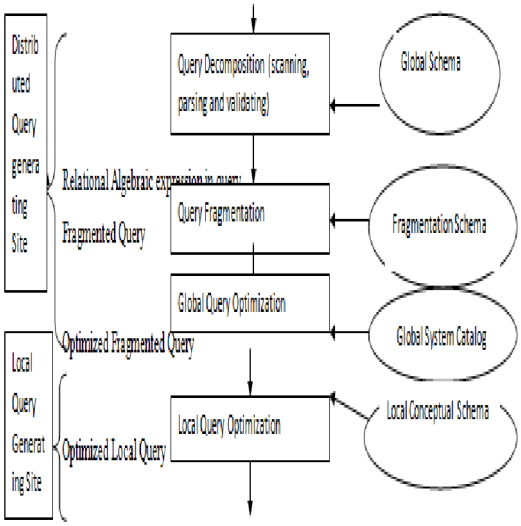

Fig. 2 optimization model in distributed databases.

III. OPTIMIZATION OF VARIOUS QUERIES:

A. Joins or Semi-Joins: Semi-Joins should be preferred over Joins. It would take less CPU, I/O cost and memory as well.

For Example: Lets consider an operator tree to extract name of employees having Manager no: 373 and salary less than 35000.

General operator Tree:

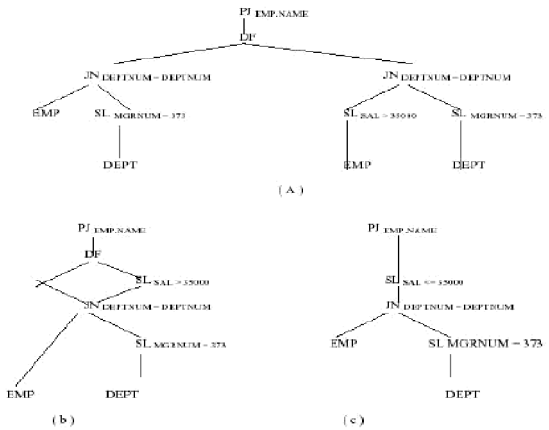

Fig. 3 General operator tree

Progressive Simplification of Query Operator tree

Cost Analysis:

--First Method:

select * intojoin2 fromemp where exists(select * fromdept1 wheredept1.mngrno=373 andemp.salary>35000 anddept1.deptno=emp.deptno);

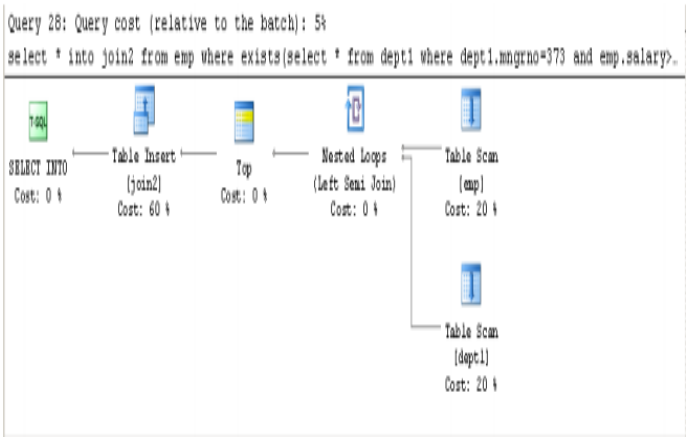

Fig. 4 shows cost of various operations using Joins and semi joins

Query Cost: 5%

Table Scan Cost: 20%

B. Second Method

selectempname fromemp where exists(select * fromdept1 wheremngrno=373 and emp.deptno=dept1.deptno)

Except

selectempname fromemp where exists(select * fromdept1 wheredept1.mngrno=373and emp.salary>35000 and dept1.deptno=emp.deptno);

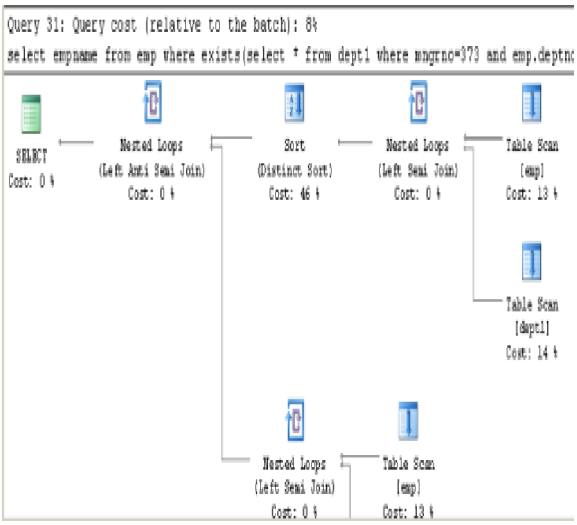

Fig. 5 shows cost of various operations using binary operators

Query Cost: 8%

Table Scan Cost: 13-14%

C. Optimized Query

SELECT empname fromemp e,dept1 d wheree.salary<=35000 andd.mngrno=373 and e.deptno=d.deptno;

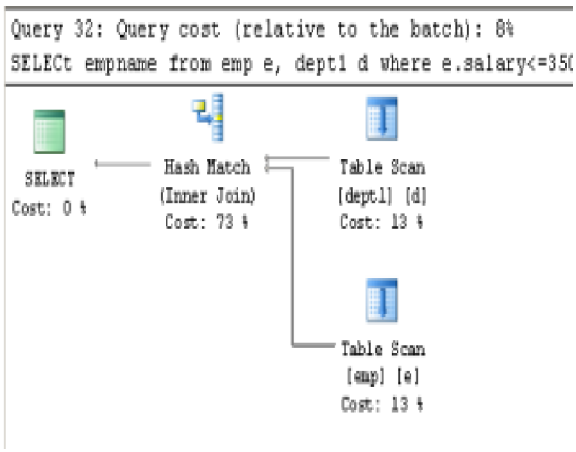

Fig. 6 shows cost of various operations using optimization.

Query Cost: 8%

Table Scan Cost: 13%

CONCLUSION:

Success of a database resides with work of its Query Optimizer, its quality and functionality. In spite of lots more research going on, still there are many issues in Query optimization. Query optimization improves time, cost and efforts. In spite of less growth, it's an exciting field.

REFERENCES:

[1] Apers, P.M.G., Hevner, A.R., Yao, S.B. Optimization Algorithms for Distributed Queries. IEEE Transactions on Software Engineering, Vol 9:1, 1983.

[2] Bancilhon, F., Maier, D., Sagiv, Y., Ullman, J.D. Magic sets and other strange ways to execute logic programs. In Proc. of ACM PODS, 1986.

[3] Bernstein, P.A., Goodman, N., Wong, E., Reeve, C.L, Rothnie,J. Query Processing in a System for Distributed Databases (SDD-1), ACM TODS 6:4 (Dec 1981).

[4] Chaudhuri, S., Shim K. An Overview of Cost-based Optimization of Queries with Aggregates. IEEE DE Bulletin, Sep. 1995. (Special Issue on Query Processing).

[5][JPlod00] J. Płodzień, A. Kraken, "Object Query Optimization through Detecting Independent Subqueries", Information Systems, Elsevier Science, 25(8), 2000, pp. 467-490.

[6][Mich09] Michel Bleja, Krzysztof Stencel, Kazimierz Subeita, Optimization of Object-Oriented Queries Addressing Large and Small Collections", Proc. Of the IMCSIT, 2009, ISBN 978-83-60810-22-4, Vol.4, pp. 643-680.

[7][Subi95] K.Subieta, C.Beeri, F.Matthes, J.W.Schmidt. "A Stack-Based Approach to Query Languages". Proc.2nd East-West Database Workshop, 1994, Springer Workshops in Computing, 1995, 159-180.

[8][MA05]Minyar Sassi, and Amel Grissa-Touzi "Contribution to the Query Optimization in the Object-Oriented Databases" World Academy of Science, Engineering and Technology 11 2005

[9][G99]G. Gardarin, "Object and relational databases" , Eyrolles, 1999.

[10] [RC 94]R.G.G. Catell, "Object-Oriented Data Management: Object-Oriented and Extended Relational Database Sytems", Addison-WesleyPublishing, Inc., 1994.

[11] Ganski, R.A., Long, H.K.T. Optimization of Nested SQLQueries Revisited. In Proc. of ACM SIGMOD, San Francisco,1987.

[12] Gassner, P., Lohman, G., Schiefer, K.B. Query Optimization in the IBM DB2 Family. IEEE Data Engineering Bulletin, Dec. 1993.

