Database Query Optimization Process

Query optimization involves a wide range of strategies applied between the input of the queries and the subsequent production of query results. They include modification of the queries and efficient retrieval of the query results. Moreover, they may generate different access methodologies and choose the optimum one to produce the requested data. Query optimization translates the non-procedural query specification into a procedural one. A plan is generated called *a query execution plan* (QEP). Different alternative QEPs for the same query are explored and the best QEP is chosen by query optimization. In [Frey89], three aspects are defined that characterize any optimization process : *the QEP generation*, *the search strategy* and *the cost function*. The optimization process is illustrated in Figure 1.

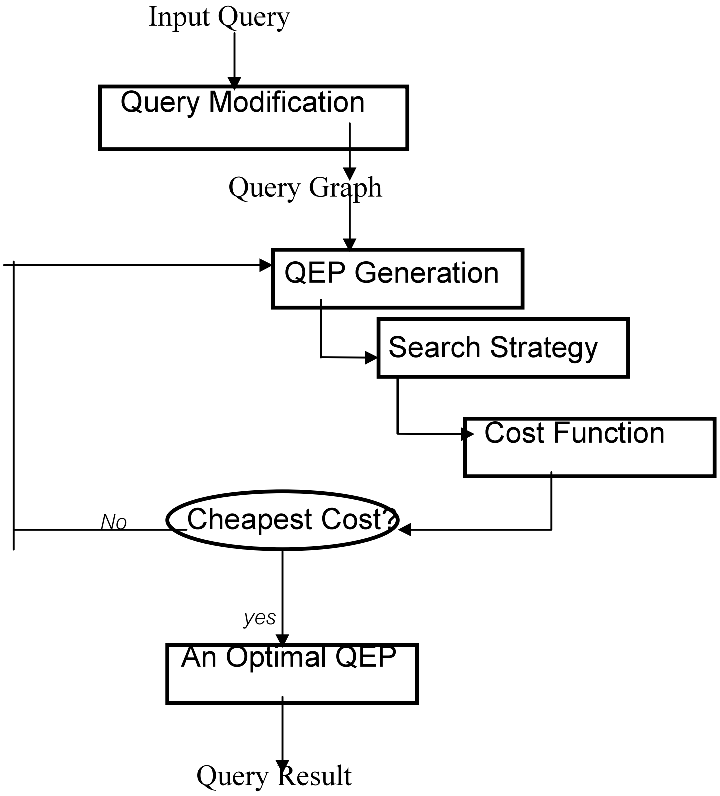

Figure 1 Query Optimization Process

Queries

A query is defined as a language expression for the retrieval of data held in databases [Jark84]. A query can be used to define the semantics of an application (functions requested by the users) or it can be used to define the function which is performed by a program which accesses a database.

Query Modification

Query modification is a stage that rewrites the initial query in order to improve efficiency during the evaluation of the query. It transforms an expression of internal representation into an equivalent form by using well-defined rules which specify the transformation. The input query can be represented by either relational algebra [Codd72] or a graph, called *a query graph* or *a connection graph* [Ullm82].

QEP Generation

QEP generation defines how to create all possible QEPs from a query graph. The set of available operators that manipulate sets of tuples once they are retrieved from stored relations create a QEP.

Search Strategy

Since a user-submitted query can be evaluated by many possible QEPs, it is necessary to locate the best plan which gives the optimal cost from among the different QEPs.

Cost Function

A cost function provides the basis for comparing different QEPs and for choosing the best plan for execution. The search strategy performs various operations based on the calculated cost values and finally leads to one QEP being selected to be the anticipated best for evaluation.

The total cost of a distributed execution plan can be expressed in terms of either the *response time* measure [Epst78] or the *total cost* measure [Sacc82]. Costs are generally specified in terms of time unit.

- The response time is the time from the initiation of the query to the time when the answer is produced.
- The total cost is the sum of the local processing cost and the communication cost. The local processing costs are usually evaluated in terms of the number of disk accesses and the central processing unit (CPU) processing time, while communication costs are described in terms of the total amount of data transmitted. If no relation is fragmented, then the local processing cost only is involved, i.e., the projections and selections are performed. However, when joins are executed, communication costs between different sites must be measured, in addition to the local processing cost. For example, if R_1 and R_2 are in different sites, R_1 must be sent to the site where R_2 is located. Or alternatively, R_2 must be sent to the site of R_1 before the operation is performed. The total cost to be minimized is the sum of the following [Ozsu91]:

$$Total\ Cost = C_{CPU} \times \#insts + C_{I/O} \times \#I/Os + C_{MSG} \times \#msgs + C_{TR} \times \#bytes$$

#insts be defined as the number of program instructions.
#I/Os is defined as the number of transfers to or from disk.
#msgs is defined as the number of messages transferred between one site and another.
#bytes is defined as the total number of data sizes transmitted in all messages.
C_{MSG} is the fixed cost of initiating and receiving a message.

Communication Cost (*C_{TR}*). The communication cost is the cost of transmitting data between sites participating in the execution of the query. This cost is incurred in processing the messages and in exchanging the data on the communication network.

Secondary Storage Access (*$C_{I/O}$*). The secondary storage cost is the cost of loading data pages from secondary storage into main memory. The cost depends on the volume of data to be retrieved, the disk page size, the available buffer size and the speed of the devices.

Computation Cost (*C_{CPU}*). The computation cost is the cost of using the central processing unit (CPU).

Query Optimization.

The cost function assigns a cost to each QEP. An *optimal QEP* is the best of all searched plans in terms of the cost function. The optimal QEP is the plan that produces the cheapest cost. Query optimization is defined as the problem of finding the most efficient query execution plan (QEP) for a query expression.

Reference

[Bern81] Bernstein P.A., Wong E., Reeve C.L., and Rothnie Jr.J.B., "Query Processing in a System for Distributed Databases (SDD-1)", *ACM Transaction on Database Systems*, Vol.6, No.4, December, 1981, pp.602-625.

[Codd72] Codd E.F., "Relational Completeness of Data Base Sublanguage", *Courant Computer Science Symposia*, No.6: Data Base Systems, R. Rustin, Ed., Prentice-Hall, 1972. pp.65-98.

[Epst78] Epstein R., Stonebraker M., and Wong E., “Query Processing in a Distributed Relational Database System”, *In Proc. ACM SIGMOD Int. Conf. On Management of Data*, May 1978, pp.169-180.

[Frey89] Freytag J.C., "The Basic Principles of Query Optimization in Relational Database Management Systems", *Information Processing 89*, G.X. Ritter (ed.), Elsevier Science Publishers B.V. (North-Holland), pp.801-807.

[Jark84] Jarke M. and Koch J, "Query Optimization in Database Systems", *ACM Computing Surveys*, Vol.16, No.2, June 1984, pp.111-152.

[Ozsu91] Ozsu M.T., and Valduriez P., *Principles of Distributed Database Systems*, Prentice-Hall, New Jersey, 1991.

[Sacc82] Sacco M.S., and Yao S.B., “Query Optimization in Distributed Database Systems”, *In M.C. Yovits (ed.), Advances in Computers*, Vol.21, New York Academic Press, 1982, pp.225-273.

[Ullm82] Ullman J.D., "Principles of Database Systems", 2nd edition, Rockville, Md.: Computer Science Press, 1982.