



Analysis of Cost Estimation used in Query Optimization of Fuzzy Relational Databases based on Sort-Merge Algorithm

S. DEEPA

Assistant Professor, DoS in Computer Applications, Pooja Bhagavat Memorial Mahajana Education Centre, KRS Road, Metagalli, Mysuru – 570016, Karnataka, India.

(Received: February 16, 2015; Accepted: April 18, 2015)

ABSTRACT

Query optimization in fuzzy relational databases aims to come out with a minimal execution cost for the available execution strategies. Each type of cost is estimated by a cost function. All cost functions together with their parameters and assumptions forms a cost model for the fuzzy query optimizer. The cost function usually takes the size of tables as inputs. It is possible that exact information is not available where by fuzzy data is assumed. Also estimating the nature of different cost models needs to be examined. A fuzzy cost function is used for this purpose which produces a fuzzy value that represents a soft estimate of the real cost while a traditional crisp cost function, produces a hard crisp estimate.

Key words: Fuzzy data, Query optimization, Cost function.

INTRODUCTION

The cost based query optimization technique searches the solution space to a problem for a solution that minimizes an objective (cost) function. The cost functions used in query optimization are estimates and not exact cost functions. So the optimization may select a query execution strategy that is not the optimal one. Evaluation of query execution is based on various components like access cost, storage cost, computation cost, memory usage cost and communication cost. To estimate the costs of various execution strategies, we must keep track of any information that is needed for the cost functions.

The query optimizer usually performs the process of fuzzy cost estimation by first finding the different search spaces and then making the cost estimation using an enumeration algorithm.

This analysis is based on cost enumeration that uses sort merge algorithm.

Query optimization architecture

The main objective of query optimization is to select the optimal execution strategy for the execution of queries. Relational query languages provide a high-level “declarative” interface to access data stored in relational databases. Over time, query language has emerged as the standard for relational query languages. Two key components of the query

evaluation component of a database system are the query optimizer and the query execution engine.

The query execution engine implements a set of physical operators. An operator takes as input one or more data streams and produces an output data stream. Examples of physical operators are (external) sort, sequential scan, index scan, nested loop join, and sort-merge join. The simplest way to think of physical operators is as pieces of code that are used as building blocks to make possible the execution of SQL queries. An abstract representation of such an execution is a physical operator tree, as illustrated in Figure 2.1.

As common queries are expressed as a sequence of selection, projection and join this optimization framework evaluates the query based on the number of joins in the input query.

Based on this operator trees are constructed for various execution strategies. Operator trees are estimated based on the various performance criteria like secondary storage usage, memory usage and communication cost. As the framework deals with fuzzy data the problem of fuzzy query optimization is to find a strategy with the least fuzzy cost. The least fuzzy cost is derived

from the comparisons among the estimated fuzzy costs. The different fuzzy costs represent different optimization criteria. The basic methods considered for determining fuzzy parameters are constructing a fuzzy parameter on the basis of experiments, constructing a fuzzy parameter on the basis of external characteristics and improving the fuzzy parameter on the basis of run time information.

Cost estimation

The estimation of the least fuzzy cost will be based on the ordering between two fuzzy costs. There are several ways to define the ordering. One simple way is by defining the grade of membership in the fuzzy costs.

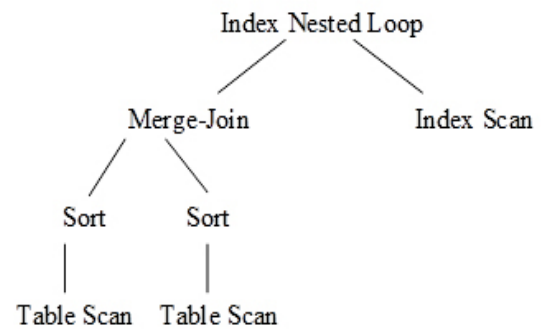


Fig. 1: Operator Tree

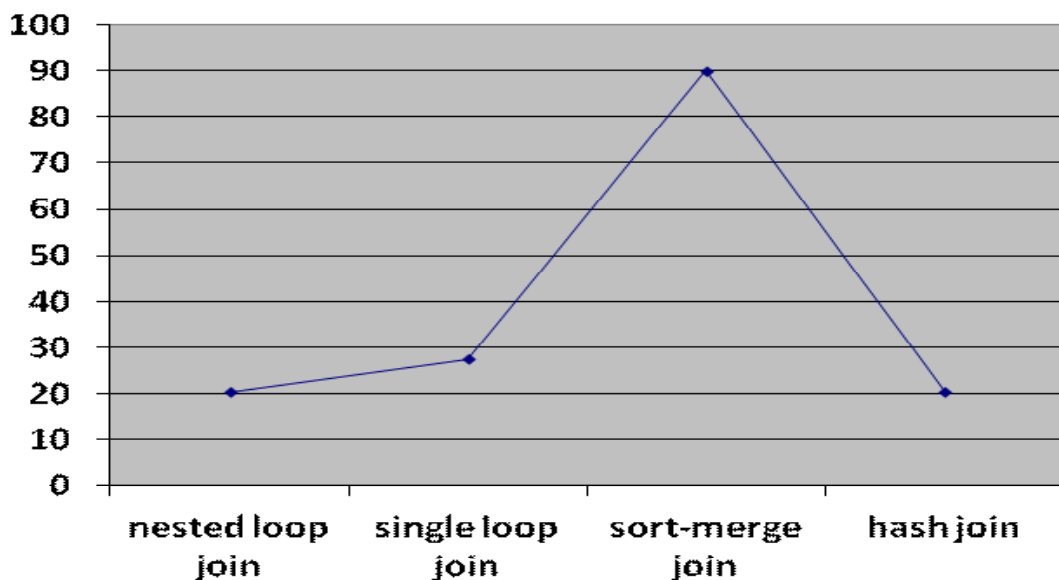


Fig. 3.1: Execution strategy based on optimization time

Fuzzy query optimization framework needs to provide a search space and cost estimation technique that can be used to assess the most optimal execution strategy. The basic operations like selection, projection and join are examined for different execution strategies.

search, binary search, hash key, primary key search and clustering index search., joins can be implemented using nested loop join, single loop join, sort-merge join and hash join. Based on these various execution strategies a sort merge algorithm was framed to find out the best execution

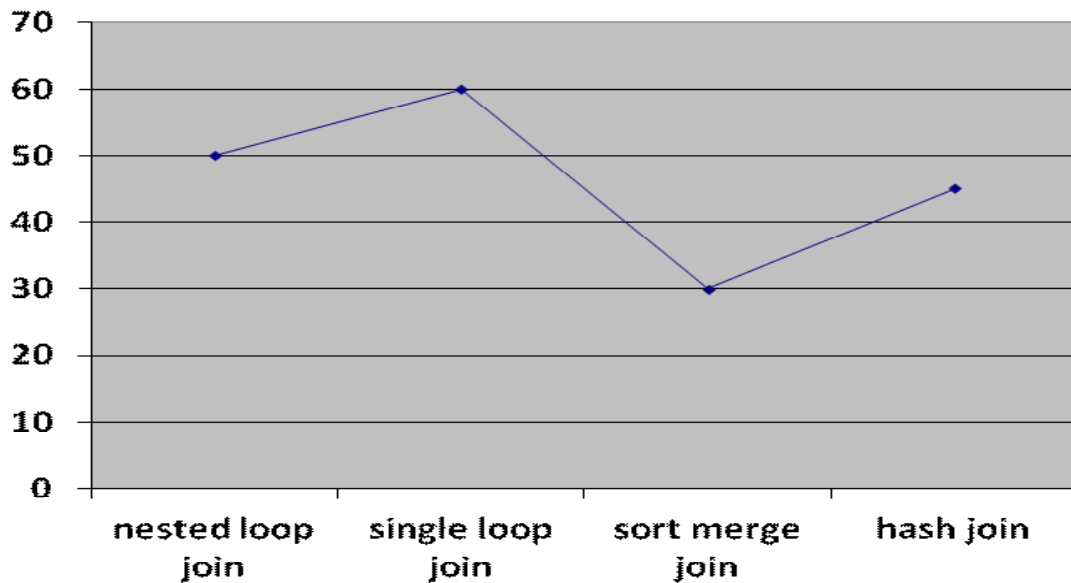


Fig. 3.2: Execution strategy based on estimated execution cost

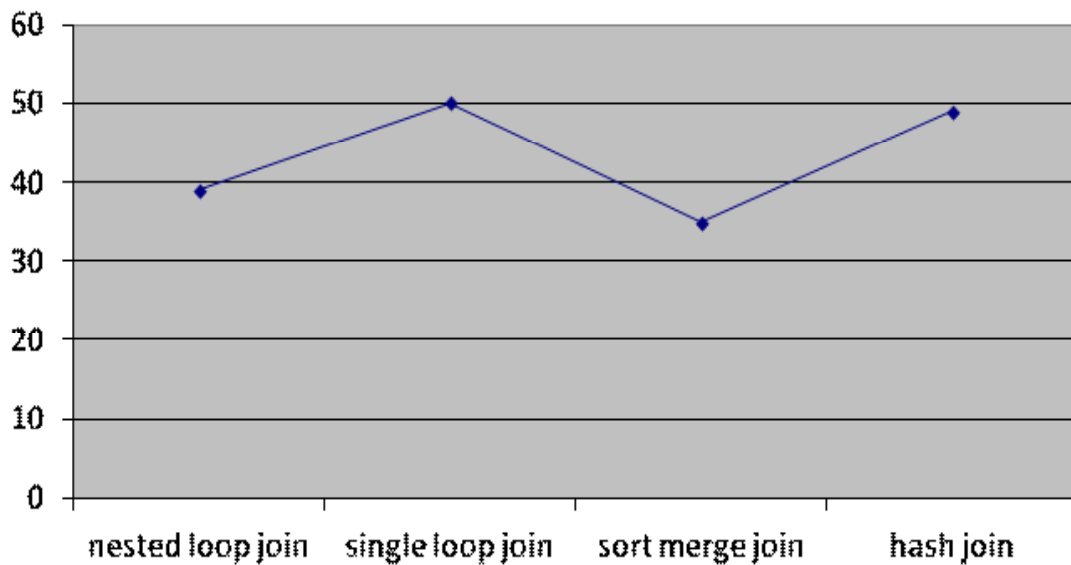


Fig. 3.3: Execution strategy based on memory requirements

strategy. As joins are based on different strategy a statistical analysis was performed based on the various performance criteria. The performance criteria considered was based on optimization time, estimated execution costs and memory requirements.

Optimization in fuzzy relational databases

Figure 3.1, 3.2 and 3.3 shows the effect of the optimization in the fuzzy relational optimizer. The increase in the optimizer time is modest while the improvement in the estimated cost is as high as 50%. It requires a significant effort to incorporate this optimization into an optimizer but the benefits can be significant if the query involves multiple joins. Incorporating this optimization into an optimizer is recommended if the system is likely to have a lot of complex decision support.

From the performance criteria's it has been observed when the number of join increases the sort merge strategy is the best. Based on this performance criterion a cost estimate is made with respect to the fuzzy information and finally the enumeration algorithm plays the role in picking up the optimal strategy.

RESULTS

Applying the fuzzy cost functions in the fuzzy cost model we can derive a fuzzy cost

estimate of an execution strategy for a global query. Fuzzy parameters can be determined either by constructing fuzzy parameter based on experiments or by improving a fuzzy parameter on the basis of runtime information. Fuzzy cost model is usually a generalization of a conventional crisp cost model. The parameters of a crisp cost model can be considered as experts crisp choices. A fuzzy cost model allows the experts to describe their fuzzy perception which appears closer to the real world than a forced crisp cost model.

CONCLUSION

In a fuzzy cost model each type of cost is estimated by a cost function. All cost function together with their parameters and assumptions forms a cost model for fuzzy relational databases. Using the cost model the optimizer can choose a good execution strategy by comparing the cost of alternative strategies.

Performance information is usually reflected in the coefficients of cost functions. However, such performance information may not be accurately known by the optimizer. In this case, fuzzy coefficients can be used in cost functions to allow imprecise information.

REFERENCES

1. Johann Christoph Freytag , A Rule Based View of Query Optimizer, In Proceedings of the 1987 ACM – SIGMOD Conference, San Francisco, California, May 1987.
2. G. Graefe and D.J.Dewilt, The Exodus Optimizer Generator, In Proceedings of the 1987 ACM – SIGMOD Conference, San Francisco, California, May 1987.
3. G. Graefe and W.J.McKenna, The Volcano Optimizer: Extensibility and Efficient Search, In Proc. IEEE Conf. on Data Eng., Vienna, Austria 1993.
4. Goetz Graefe, The Cascades Framework for Query Optimization, *In Bulletin of the Technical Committee on Data Engineering*, **18**; 19-29; (1995).
5. S. Deepa , A Multivalued Dependency-Based Normalization Approach for Symbolic Relational Databases, *The IUP Journal of Computer Sciences*, **5**(3); 40 - 46 (2011).
6. S. Deepa , A Query Optimization Framework for Fuzzy Relational Databases, *Asian Journal of Engineering and Applied Technology*, **1**(1): 43 – 16 (2012).
7. Bipin C. Desai, An Introduction to Database Systems, Galgotia Publishing Pvt. Ltd., pp. 460-486.
8. Goerge J. Klir, Tina A. Folger, Fuzzy sets, Uncertainty and Information, Prentice – Hall of India Pvt. Ltd., pp.138 -140.