Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol. 1 No. 1, 2012, pp.43-46

DOI: https://doi.org/10.51983/ajeat-2012.1.1.2502

A Query Optimization Framework for Fuzzy Relational Databases

S. Deepa

Department of Studies in Computer Science,

PoojaBhagavat Memorial Mahajana PG Centre, Metagalli, Mysore - 570 016, Karnataka, India

E-mail : dishan9969@gmail.com

(Received on 06 January 2012 and accepted on 15th March 2012)

***Abstract* - Ever since the development of relational model, relational database systems have been extensively studied and several commercial relational database systems are currently available. Relational model usually take care of only well defined data. In order to capture more meaning to the data an extension of the classical relational model called fuzzy relational model was proposed. The key reasons for the success of relational database lies in the power of declarative languages and execution strategies used in query optimization. Estimating the cost of fuzzy query based on system catalog introduces error due to approximation involved and insufficient information at query execution time. So there is need for a query optimization framework that addresses the issues of query execution in fuzzy relational databases. This paper deals with a framework for building fuzzy cost model to obtain a good execution strategy for a query.**

***Keywords :* Query Optimization, Fuzzy Relational Databases, Fuzzy Cost**

I. Introduction

Query optimization usually seeks an optimal execution strategy for a query. The criteria for a good strategy have no well defined boundary. The criteria's used in fuzzy database environment need to be tolerant of imprecise information. The major distinction between query optimization in fuzzy and crisp database models lies in the mathematical model used to evaluate the execution cost of a query.

Evaluation involved in the execution cost of a query is based on the information maintained in the system catalog. This may introduce errors in the estimates if the statistics are not kept up-to-date. The problem is further aggravated in the case of fuzzy data base systems where we deal with imprecise data. Hence there is a need for an optimization framework, to provide solution to the query optimizers which deal with imprecise data. The perspective of this paper is to come out with a fuzzy cost model for selecting a good or not bad execution strategy. The Section 2 of the paper states the fuzzy optimization problem and includes the cost model, Section 2 focuses on the analysis and interpretations of the fuzzy cost model including the discussion of results and Section 4 concludes with the summary of the paper.

II. Fuzzy Optimization Problem

The query optimization in a fuzzy database environment can be stated as:

For a given query, there are usually a number of feasible execution strategies. Let Y = $\{y_1, y_2, \ldots\ldots y_n\}$ be the set of all execution strategies for a given query Q. For each strategy $y_i(1<i<n)$, a fuzzy cost estimate c_i can be calculated. Let $C=\{c_1,c_2,\ldots c_n\}$[6]. The fuzzy cost model actually induces a fuzzy function between the strategies involved and fuzzy costs as

F :Ya C. The problem of fuzzy query optimization is to find a strategy $y_0 \varepsilon$ Y with the least fuzzy cost, that is,

$F(y_0)$="min" F(y) (1)

y ε Y

A. Fuzzy Cost Model

Query optimizers for fuzzy relational databases evaluate the performance of a query by using fuzzy coefficients that allows imprecise information.The cost function for fuzzy databases considers the size of table as inputs including the effects of implementation, optimization, and query execution plans produced [1].The fuzzy cost model evaluates a query based on various execution strategies like selection, projection and join. The fuzzy function based on the size of the relation and various performance factors can be stated as

$FC(|R|)=D_i(R)+D_i(R)+\ldots\ldots..$ (2

i = 1,2……n

Where FC is the fuzzy cost, R is the cardinality of the relation and Di is the performance parameters based on the functions like retrieving a tuple from the relation R and processing it.

Fuzzy optimization aims at estimating a minimum cost based on evaluation made by the cost model. Cost model depends on various factors like cardinality of the relation, CPU and I/O cost for query execution. The cost factors in the

cost model are enumerated by using the sort-merge algorithm [3]. In the sorting phase, runs of the file that can fit in the available buffer space are read into main memory and then sorted. In the merging phase, the sorted runs are merged during one or more passes.

The outline of the algorithm is :

Set I ← 1

J ← b [size of the file in blocks]

K← n_B[size of buffer in blocks]

M ← [(J/K)]

sort phase}

While (I < M)

Do {

Read next blocks of the file into the buffer or if there are less than k blocks remaining, then read in remaining blocks;

Sort the records in the buffer and write as a temporary subfile;

I← I + 1;

}

merge phase: merge subfiles until only one remains}

III. Analysis And Interpretations

The fuzzy relational query optimizer picks up an optimal strategy based on the number of joins in the input query[2]. Based on the count of joins different execution plans will be evaluated to form the search space of optimization. The search space implements the execution strategies in the form of operator trees.Taking a sample fuzzy query as below:

Select pnumber, dnum, lname, address,bdate

from project, department, employee

Where dnum=dnumber and mgr_ssn=ssn

andplocation='stafford';

In the above query mgr and plocation are assumed to be fuzzy values as it is supposed to take multiple values. As the fuzzy relational query optimization picks up an optimal strategy based on the number of joins in the input query so, for the above query different execution plans based on the different joins forms the search space. The search space that implements various execution strategies are shown below as operator trees.

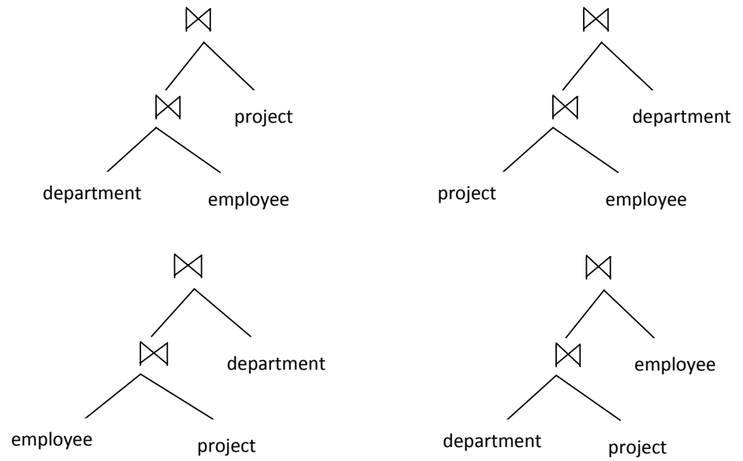

Fig. 1 Operator Tree

A. Fuzzy Cost Estimation

Based on the different execution strategies as shown in the form of operator trees in figure1, a statistical analysis is performed to evaluate the fuzzy cost based on various parameters mentioned in the equation (2). The table below shows different parametric values for the query mentioned above

TABLE I STATISTICAL INFORMATION FOR RELATION PROJECT

Parameter	Fuzzy estimate
S11	{0.7/2.8,0.8/3.5}
|R1|*t1	{0.7/78.1,0.9/75.2}
|R12|*t1	{0.4/46.3,0.8/65.8}

TABLE II STATISTICAL INFORMATION FOR RELATION DEPARTMENT

Parameter	Fuzzy estimate
S12	{0.5/52.9,0.7/359.9}
|R2|*t2	{0.3/0.07,0.1/0.3}
|R12|*t2	{0.5/5.8,0.3/0.02}

TABLE III STATISTICAL INFORMATION FOR RELATION EMPLOYEE

Parameter	Fuzzy estimate
S13	{0.2/2.3,0.4/5.8}
|R3|*t3	{0.3/0.02,0.5/0.1}
|R13|*t3	{0.7/3.0,0.4/394.1}

As mentioned in the operator tree, the first join is between department and employee. The second join is between department, employee and project. Both the join and access method foe the input relation must be determined. Since department has no index the only available access method is linear search. The project relation will have the selection operation performed before the join, so two options exist: linear search or index scan. So the cost estimation will be based on comparison.

The index is non-unique in the join so the optimizer assumes a normal data distribution and eliminates the number of record pointers. This is computed from the parameter values specified in the table 1, 2 and 3 by estimating the cost of using the index and accessing the records to be block accesses.

The file size is calculated by determining the blocking factor and rows per block. The ordering for optimization is based on the grades of membership. To take into consideration the elements with lower grades of membership in a fuzzy cost the ordering between two fuzzy costs is defined as:

$$\omega (\text{“min”}\ F(y))\ d\text{”}\ \omega (F(x)),\ V\ x \varepsilon Y \quad (3)$$

$$y \varepsilon Y$$

where ω is the weighted average value of a fuzzy cost and

$$\omega = \frac{1}{W} \sum_{i=1}^{n} x_i \quad (4)$$

For the query specified in Section 4.5 let c_1,c_2,c_3 and c_4 be the different execution strategies.

$C_1 = \{0.4/46.3, 0.8/65.8\}$

$C_2 = \{0.7/3.0, 0.4/394.1\}$

$C_3 = \{0.5/5.8, 0.3/0.02\}$

$$\omega (c_1) = \frac{1}{1.9}(0.4 * 46.3 + 0.8 * 65.8) = 8.2$$

$$\omega (c_2) = \frac{1}{2.5}(0.7 * 3.0 + 0.4 * 394.1) = 63.14$$

$$\omega (c_3) = \frac{1}{1.6}(0.5 * 5.8 + 2.9 * 0.006) = 1.81625$$

The fuzzy criterion induces a fuzzy set where by the choice is to select the minimal cost. The minimal cost is given by c_3 and this gives the optimal strategy. Thus a fuzzy parameter estimate reflects the fuzzy perception for the database. Errors made in a forced crisp parameter estimate can be adjusted by the values with lower grades of membership in the fuzzy parameter estimate. Hence a fuzzy approach has a higher chance to be successful in choosing a good execution strategy for a query than a crisp one when a correct crisp choice is difficult to obtain.

B. Results

The elements with higher grades of membership in a fuzzy set are more possible to belong to the set than those with lower grades of membership in the fuzzy set. That is the reason why smaller weights are given to the elements with lower grades of membership when the weighted average value is calculated for the fuzzy value. Based on this property, a fuzzy value in the fuzzy cost model can be approximated by another fuzzy value that is obtained by removing some elements with lower grades of membership. Thus the cost model with the smallest grade of membership is chosen as the best execution strategy. Thus the objective of minimal cost estimation is done.

III. CONCLUSION

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 framework uses an enumeration algorithm based on sort merge strategy. 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. Further exploration of fuzzy cost model based on different execution strategies can be considered. Feasibility of heuristic based fuzzy optimization forms the scope of future research. Query execution based on heuristics may lead to different execution strategies. Heuristics can be used to reduce the number of plans considered, and thereby reduce the cost of optimization.

REFERENCES

[1] Abraham Silberschatz, Henry F Korth and Sudarshan S (2006), Database System Concepts, 5th Ed., Mc-Graw Hill International, pp. 300-350.

[2] Bipin C. Desai, An Introduction to Database Systems, Galgotia Publishing Pvt. Ltd.,, pp. 460-486.

[3] B. Buckles and F. Petry, “A Fuzzy model for relational Databases”, *Fuzzy Sets and Systems*, Vol.7, 1982, pp 213-226.

[4] S. Deepa “A Multivalued Dependency-Based Normalization Approach for Symbolic Relational Databases, *The IUP journal of Computer Sciences,* Vol. 5, No. 3, 2011, pp 40-46.

[5] GoetzGraefe “ The Cascades Framework for Query Optimization”. *In Bulletin of the Technical Committee on Data Engineering,* Vol. 18, No. 3, pp. 19 – 29, September 1995.

[6] Qiang Zhu and P.A. Larson, “Building regression cost models for MultidatabaseSytems”, *Proc. 4 th IEEE Int. Conf. on Parallel Distributed Information Systems, Miami Beach, Florida,* Dec. 1996, pp. 220-231.

[7] S. Pak et al., “Fuzzy Querying in Relational Databases”, *Proc. 5th IFSA world Congress,* 1993, pp. 553 – 536.

[8] Qiang Zhu and P.A. Larson, “Building regression cost models for multidatabase sytems”, *Proc. 4 th IEEE Int. Conf. on Parallel Distributed Information Systems, Miami Beach, Florida,* Dec. 1996, pp. 220-231.

[9] Navin Kabra, “Query Optimization for Object-Relational Database Systems”, Ph D Thesis,University of Wisconsin –Madison, 1999.