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 : disha996@gmail.com

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, ..., y_n} be the set of all execution strategies for a given query Q. For each strategy y_i, a fuzzy cost estimate c_i can be calculated. Let C={c_1,c_2,...,c_n}[6]. The fuzzy cost model actually induces a fuzzy function between the strategies involved and fuzzy costs as

\[ F: Y \rightarrow \mathbb{R} \]

\[ F(y) = \min_{y \in Y} F(y) \]  \hspace{1cm} (1)

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

\[ FC(R) = D(R) + D(R) + ... + D(R) \]  \hspace{1cm} (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 \hspace{1cm} \text{[size of the file in blocks]}
K ← n \hspace{1cm} \text{[size of buffer in blocks]}
M + ← \text{[(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;
1e ← 1 + 1;
} merge phase: merge subfiles until only one remains]
A Query Optimization Framework for Fuzzy Relational Databases

A. Fuzzy Cost Estimation

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:
\[ \alpha (\{ \min \} F(y)) d \alpha (F(x)), \forall x \neq y \] (3)
where \( \alpha \) is the weighted average value of a fuzzy cost and
\[ \alpha = \frac{1}{n} \sum_{i=1}^{n} x_i \]
For the query specified in Section 4.5 let \( c_1, c_2, c_3 \) and \( c_4 \) be the different execution strategies.
\[ C_1 = [0.4/46.3, 0.8/65.8] \]
\[ C_2 = [0.7/3.0, 0.4/394.1] \]
\[ C_3 = [0.5/5.8, 3.0/0.02] \]
\[ C_4 = [1.0/4.63, 0.8/65.8] \]
\[ \omega (c_i) = \frac{1}{n} (0.4 * 46.3 + 0.8 * 65.8) = 8.2 \]
\[ \omega (c_1) = 1.9 \]
\[ \omega (c_2) = 63.14 \]
\[ \omega (c_3) = 2.5 \]
\[ \omega (c_4) = 1.81625 \]
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.

Table I Statistical information for relation project

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fuzzy estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11</td>
<td>(0.5/2.0, 0.3/3.5)</td>
</tr>
<tr>
<td>B[P1]</td>
<td>(0.5/7.8, 0.3/7.5)</td>
</tr>
<tr>
<td>B[1]P1</td>
<td>(0.4/43.2, 8.5/8)</td>
</tr>
</tbody>
</table>

Table II Statistical information for relation department

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fuzzy estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>S12</td>
<td>(0.5/2.0, 0.7/59.9)</td>
</tr>
<tr>
<td>E2[P2]</td>
<td>(0.3/0.07/0.16/3)</td>
</tr>
<tr>
<td>E[2]P2</td>
<td>(0.5/5.0, 3.0/0.02)</td>
</tr>
</tbody>
</table>

Table III Statistical information for relation employee

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fuzzy estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>S13</td>
<td>(0.2/2.0, 0.4/5.8)</td>
</tr>
<tr>
<td>B3[P3]</td>
<td>(0.3/0.02/0.5/8)</td>
</tr>
<tr>
<td>B[3]P3</td>
<td>(0.7/3.0/0.4/594.1)</td>
</tr>
</tbody>
</table>

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 for 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

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