CSE 562
Database Systems

Query Processing: Overview

Some slides are based or modified from originals by
Database Systems: The Complete Book,
Pearson Prentice Hall 2nd Edition
©2008 Garcia-Molina, Ullman, and Widom

cse@buffalo

Outline – Query Optimization

• Overview
• Relational algebra level
  – Algebraic Transformations
• Detailed query plan level
  – Estimate Costs
    – Estimating size of results
    – Estimating # of IOs
  – Generate and compare plans

Query Processing

• The query processor turns user queries and data
  modification commands into a query plan – a
  sequence of operations (or algorithms) on the
  database
  – from high level queries to low level commands
• Decisions taken by the query processor:
  – Which of the algebraically equivalent forms of a query
    will lead to the most efficient algorithm?
  – For each algebraic operator, what algorithm should we
    use to run the operator?
  – How should the operators pass data from one to the
    other? (e.g., main memory buffers, disk buffers)

Example

SELECT R.B, S.D
FROM R, S
WHERE R.A = 'c' AND S.E = 2 AND R.C = S.C
Example (cont.)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>10 x 2</td>
<td>b</td>
<td>1</td>
<td>20</td>
<td>20 y 2</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
<td></td>
<td>c</td>
<td>2</td>
<td>10</td>
<td>30 z 2</td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td></td>
<td>d</td>
<td>2</td>
<td>35</td>
<td>40 x 1</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
<td></td>
<td>e</td>
<td>3</td>
<td>45</td>
<td>50 y 3</td>
</tr>
</tbody>
</table>

Answer

<table>
<thead>
<tr>
<th>B</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>x</td>
</tr>
</tbody>
</table>

Query Processing

How do we execute a query eventually?

- Scan relations
- Do Cartesian product
- Select tuples
- Do projection

One idea

Example (cont.)

<table>
<thead>
<tr>
<th>R x S</th>
<th>R.A</th>
<th>R.B</th>
<th>R.C</th>
<th>S.C</th>
<th>S.D</th>
<th>S.E</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>y</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Relational algebra can be enhanced to describe plans

Example: Plan I

1. Scan R
2. For each tuple r of R, scan S
3. For each (r, s), where s in S select and project on the fly

OR: $\pi_{R.b,d}(\sigma_{R.a=c' \land S.e=2 \land R.c=S.c}(R_{SCAN} \times S_{SCAN}))$
Example (cont.)

FLY and SCAN are the defaults

Example: **Plan I**

\[
\pi_{R,B,S,D} \\
\sigma_{R.A = 'c' \text{ AND } S.E = 2 \text{ AND } R.C = S.C}
\]

Another Idea

Example: **Plan II**

Example (cont.)

Another Idea

Example: **Plan III**

Use R.A and S.C Indexes

1. Scan R and S
2. Perform on the fly selections
3. Do hash join
4. Project

Example (cont.)

Example: **Plan III**

Use R.A and S.C Indexes

1. Use R.A index to select R tuples with R.A = “c”
2. For each R.C value found, use S.C index to find matching join tuples
3. Eliminate join tuples for which S.E ≠ 2
4. Project B,D attributes
Example (cont.)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

A='c'  
<10,x,2>  
check=2?  
output: <2,x>  
next tuple: <c,7,15>

Example (cont.)

Algebraic Form of Plan

\[ \pi_{R,B,C} \sigma_{S,E=2} \]

Right Index Join

From Query To Optimal Plan

- Complex process
- Algebra-based logical and physical plans
- Transformations
- Evaluation of multiple alternatives

Issues in Processing and Optimization

- Generate Plans
  - Employ efficient execution primitives for computing relational algebra operations
  - Systematically transform expressions to achieve more efficient combinations of operators
- Estimate Cost of Generated Plans
  - Statistics
- “Smart” Search of the Space of Possible Plans
  - always do the “good” transformations (relational algebra optimization)
  - prune the space (e.g., System R)
- Often the above steps are mixed
Architecture

- SQL query
- Parse
- Parse tree
- Logical query plan
- Generate/transform \( l.q.p.(s) \)
- Improved \( l.q.p(s) \)
- Estimate result sizes
- \( l.q.p.(s) + \text{sizes} \)
- Generate physical plans
- Transform \( p.q.p.(s) \)
- \( \{P_1, P_2, \ldots\} \)

Example Journey of a Query

- **SELECT** Theater
  FROM Movie M, Schedule S
  WHERE M.Title = S.Title
  AND M.Actor = "Winger"

Example Journey of a Query (cont.)

- **Summary of Logical Plan Generator**
  - 4 logical query plans created
  - Algebraic rewritings were used for producing the candidate logical query plans
  - the last one is the winner (at least, cannot be a big loser)
  - in general, multiple logical plans may "win" eventually

Example Journey of a Query (cont.)

- **Physical Plan Generator**
  - More than one plans may be generated by choosing different primitives

Example Journey of a Query (cont.)

- Physical Plan Generator
  - Choose execution primitives and data passing
  - Index on Actor and Title
  - Unsorked tables
  - Tables >> Memory

Example Journey of a Query (cont.)

- Physical Plan Generator
  - If cond refers only to S
  - Physical Plan 1

Example Journey of a Query (cont.)

- Physical Plan Generator
  - Index on Actor and Title
  - Table Schedule sorted on Title

Example Journey of a Query (cont.)

- Physical Plan Generator
  - \( \pi_{\text{S. Theater}} \)
  - \( \sigma_{\text{M.Actor = 'Winger'}} \)
  - \( \sigma_{\text{M.Title = S.Title}} \)
  - Movie M
  - Schedule S
  - Cont.
Example: Nested SQL Query

```sql
SELECT title
FROM StarsIn
WHERE starName IN
    (SELECT name
     FROM MovieStar
     WHERE birthdate LIKE '%1960')

Find the movies with stars born in 1960
```

Example: Parse Tree

Example: Generating Relational Algebra

- An expression using a two-argument σ, midway between a parse tree and relational algebra

Example: Logical Query Plan

- May consider "IN" elimination as a transformation in the query rewriter or may consider it a task of the converter
Example: Improved Logical Query Plan

Question
Push project to StarsIn?

Example: Estimate Result Sizes

Result sizes are important for selecting physical plans

Additional parameters:
memory size, result sizes...

Example: One Physical Plan

UB CSE 562
25
UB CSE 562
90
UB CSE 562
27