Access Path Selection in a Relational Database Management System

Presenter: Dutch Meyer
Discussion: Mike “Debo” DiBernardo
Somewhat based on slides from Stephen Ingram, modified by Rachel Pontinger.

Key Points
- Finding the right path
- Optimization based on estimation
- Tackling the problem space
- System R - the basis for the modern query optimizer

Find the Best path
- SELECT * FROM A,B,C WHERE A.n = B.n AND B.m = C.m
- A = 100 tuples
- B = 50 tuples
- C = 2 tuples
- Which plan is cheaper?
  - Join( C, Join( A, B ) )
  - Join( A, Join( B, C ) )

How hard is this problem?

Relation Optimization
- With a single relation, it’s not so bad
- We can consider all paths

Join Optimization
- 2-Way joins
- N-Way joins
- How many permutations of N-Way joins could there possibly be – N!?
<table>
<thead>
<tr>
<th>Optimization</th>
<th>Heuristics!</th>
</tr>
</thead>
<tbody>
<tr>
<td>- So the general case search space is big.</td>
<td>- Hard coded rules</td>
</tr>
<tr>
<td>- Possible aids:</td>
<td>- Easy to understand</td>
</tr>
<tr>
<td>- Heuristics</td>
<td>- Cheap to compile, use</td>
</tr>
<tr>
<td>- Statistics</td>
<td>- Example: Predicates</td>
</tr>
<tr>
<td>- Dynamic Programming</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Or no heuristics?</th>
<th>Statistics!</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Inflexible</td>
<td>- Generalized solution</td>
</tr>
<tr>
<td>- Non General</td>
<td>- Example: Data size, type, distribution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Or no Statistics?</th>
<th>Dynamic Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Difficult to compile</td>
<td>- Simplify search space by reusing solutions</td>
</tr>
<tr>
<td>- Difficult to store</td>
<td>- Reduce storage costs to $2^N$</td>
</tr>
<tr>
<td></td>
<td>- Not so bad, coming from N!</td>
</tr>
</tbody>
</table>
Or…

- Dynamic programming improves the effectiveness of statistics.
- Careful heuristics further limit the search space.
- Proper ordering allows the methods to work together.

Ordering

- Defn: Interesting Ordering – Orderings that we are interested in returning

Key Contributions

- Cost based optimization
  - Statistics
  - CPU utilization (for sorts, etc.)
- Dynamic programming approach
- Interesting Orders