Lecture 23: Query Optimization (3)
Monday, November 22, 2010

Outline
- Search space
- Algorithms for enumerating query plans
- Estimating the cost of a query plan

Computing the Cost of a Plan
- Collect statistical summaries of stored data
- Estimate size in a bottom-up fashion
- Estimate cost by using the estimated size

Statistics on Base Data
- Collected information for each relation
  - Number of tuples (cardinality)
  - Indexes, number of keys in the index
  - Number of physical pages, clustering info
  - Statistical information on attributes
    - Min value, max value, number distinct values
    - Histograms
  - Correlations between columns (hard)
- Collection approach: periodic, using sampling

Size Estimation Problem

S = SELECT list
  FROM R1, ..., Rn
  WHERE cond1 AND cond2 AND ... AND cond_k

Given T(R1), T(R2), ..., T(Rn)
Estimate T(S)

How can we do this? Note: doesn’t have to be exact.

Remark: T(S) ≤ T(R1) × T(R2) × ... × T(Rn)
Selectivity Factor

• Each condition $cond$ reduces the size by some factor called selectivity factor.

• Assuming independence, multiply the selectivity factors.

Example

$\begin{align*}
R(A,B) & \quad \text{SELECT} * \\
S(B,C) & \quad \text{FROM} \ R, S, T \\
T(C,D) & \quad \text{WHERE} \ R.B = S.B \text{ and } S.C = T.C \text{ and } R.A < 40
\end{align*}$

$T(R) = 30k, \ T(S) = 200k, \ T(T) = 10k$

Selectivity of $R.B = S.B$ is $1/3$

Selectivity of $S.C = T.C$ is $1/10$

Selectivity of $R.A < 40$ is $1/2$

What is the estimated size of the query output?

Rule of Thumb

• If selectivities are unknown, then: selectivity factor = $1/10$ [System R, 1979]

Selectivities from Statistics

• Condition is $A = c$ /* value selection on $R$ */
  - Selectivity = $1/V(R.A)$

• Condition is $A < c$ /* range selection on $R$ */
  - Selectivity = $(c - \text{Low}(R.A))/\text{High}(R.A) - \text{Low}(R.A))/T(R)$

• Condition is $A = B$ /* $R \bowtie A=B S$ */
  - Selectivity = $1 / \text{max}(V(R.A), V(S.A))$
  - (will explain next)

Assumptions

• Containment of values: if $V(R,A) <= V(S,B)$, then the set of A values of $R$ is included in the set of B values of $S$
  - Note: this indeed holds when $A$ is a foreign key in $R$, and $B$ is a key in $S$

• Preservation of values: for any other attribute $C$, $V(R \bowtie A=B S, C) = V(R, C)$ (or $V(S, C)$)

Selectivity of $R \bowtie A=B S$

Assume $V(R.A) <= V(S,B)$

• $\text{mmmmmmhk,mmmbknmmmnmmmmmmmmktt}$
  • Each tuple in $R$ joins with $T(S)/V(S,B)$ tuple(s) in $S$
  • Hence $T(R \bowtie A=B S) = T(R) T(S) / V(S,B)$
  • In general: $T(R \bowtie A=B S) = T(R) T(S) / \text{max}(V(R.A), V(S.B))$
Size Estimation for Join

Example:
- $T(R) = 10000$, $T(S) = 20000$
- $V(R,A) = 100$, $V(S,B) = 200$
- How large is $R \Join_{A=B} S$?

Histograms

- Statistics on data maintained by the RDBMS
- Makes size estimation much more accurate (hence, cost estimations are more accurate)

Employee($ssn$, name, age)

$T(Employee) = 25000$, $V(Employee, age) = 50$
$min(age) = 19$, $max(age) = 68$

$\sigma_{age=48}(Employee) = ?$  $\sigma_{age>28 \ and \ age<35}(Employee) = ?$

<table>
<thead>
<tr>
<th>Age</th>
<th>0..20</th>
<th>20..29</th>
<th>30..39</th>
<th>40..49</th>
<th>50..59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

Employee($ssn$, name, age)

$T(Employee) = 25000$, $V(Employee, age) = 50$
$min(age) = 19$, $max(age) = 68$

$\sigma_{age=48}(Employee) = ?$  $\sigma_{age>28 \ and \ age<35}(Employee) = ?$

<table>
<thead>
<tr>
<th>Age</th>
<th>0..20</th>
<th>20..29</th>
<th>30..39</th>
<th>40..49</th>
<th>50..59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

Estimate = $25000 / 50 = 500$
Estimate = $25000 \times 6 / 60 = 2500$

Estimate = $1200$
Estimate = $2 \times 80 + 5 \times 500 = 2660$
Types of Histograms

- How should we determine the bucket boundaries in a histogram?

Difficult Questions on Histograms

- Small number of buckets
  - Hundreds, or thousands, but not more
  - WHY?
- Not updated during database update, but recomputed periodically
  - WHY?
- Multidimensional histograms rarely used
  - WHY?

Employee(ssn, name, age)

Histograms

<table>
<thead>
<tr>
<th>Age</th>
<th>0-20</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt;60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

Eq-width:

<table>
<thead>
<tr>
<th>Age</th>
<th>0-20</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt;60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>1800</td>
<td>2000</td>
<td>2100</td>
<td>2200</td>
<td>1900</td>
<td>1800</td>
</tr>
</tbody>
</table>

Eq-depth:

<table>
<thead>
<tr>
<th>Age</th>
<th>0-20</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt;60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>1800</td>
<td>2000</td>
<td>2100</td>
<td>2200</td>
<td>1900</td>
<td>1800</td>
</tr>
</tbody>
</table>

Compressed: store separately some highly frequent values: (4K,1900)

Summary of Query Optimization

- Three parts:
  - search space, algorithms, size/cost estimation
- Ideal goal: find optimal plan. But
  - Impossible to estimate accurately
  - Impossible to search the entire space
- Goal of today’s optimizers:
  - Avoid very bad plans