CS 457: Database Management Systems

Lecture 18
Query Plan Cost Estimation
Logistics

• Midterm grades
  – To be posted today or tomorrow
  – Return papers?
    • Vote
  – Curving? (maybe, already 50 students under 80)

• HW4
  – Transactions

• Bonus point (tentative)
  – 10% towards your final grade
  – Something like:
    • Array database benchmarking (e.g., TileDB)
    • NoSQL (k-v store) implementation (e.g., Chord)
    • Distributed database prototyping
Query Optimization Summary

Goal: find a physical plan that has minimal cost

What is the cost of a plan?
For each operator, cost is function of CPU, IO, network bw

$$\text{Total\_Cost} = \text{CPU\_Cost} + w_{\text{IO}} \text{I\_Cost} + w_{\text{BW}} \text{B\_Cost}$$

Cost of plan is total for all operators
In this class, we look only at IO
Query Optimization Summary

Goal: find a physical plan that has minimal cost
Query Optimization Summary

Goal: find a physical plan that has minimal cost

Know how to compute cost if know cardinalities
Query Optimization Summary

Goal: find a physical plan that has minimal cost

Know how to compute cost if know cardinalities
Query Optimization Summary

Goal: find a physical plan that has minimal cost

Know how to compute cost if know cardinalities

– Eg. Cost($V \bowtie T$) = $3B(V) + 3B(T)$

– $B(V) = T(V) / \text{PageSize}$

– $T(V) = T(\sigma(R) \bowtie S)$
Query Optimization Summary

Goal: find a physical plan that has minimal cost

Know how to compute cost if know cardinalities

- Eg. Cost(\(V \bowtie T\)) = 3B(V) + 3B(T)
- \(B(V) = \frac{T(V)}{\text{PageSize}}\)
- \(T(V) = T(\sigma(R) \bowtie S)\)

Cardinality estimation problem: e.g. estimate \(T(\sigma(R) \bowtie S)\)
Database Statistics

- **Collect** statistical summaries of stored data

- **Estimate size** (=cardinality) in a bottom-up fashion
  - This is the most difficult part, and still inadequate in today’s query optimizers

- **Estimate cost** by using the estimated size
  - Hand-written formulas, similar to those we used for computing the cost of each physical operator
Database Statistics

- Number of tuples (cardinality) $T(R)$
- Indexes, number of keys in the index $V(R,a)$
- Number of physical pages $B(R)$
- Statistical information on attributes
  - Min value, Max value, $V(R,a)$
- Histograms

- Collection approach: periodic, using sampling
Size Estimation Problem

\[ Q = \text{SELECT} \ \text{list} \]
\[ \text{FROM} \ \ R1, \ldots, \ Rn \]
\[ \text{WHERE} \ \text{cond}_1 \ \text{AND} \ \text{cond}_2 \ \text{AND} \ldots \ \text{AND} \ \text{cond}_k \]

Given \( T(R1), \ T(R2), \ldots, \ T(Rn) \)
Estimate \( T(Q) \)

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

\[ Q = \text{SELECT list} \]
\[ \text{FROM } R1, \ldots, Rn \]
\[ \text{WHERE } \text{cond}_1 \text{ AND cond}_2 \text{ AND } \ldots \text{ AND } \text{cond}_k \]

Remark: \( T(Q) \leq T(R1) \times T(R2) \times \ldots \times T(Rn) \)
Size Estimation Problem

Q = SELECT list
   FROM R1, …, Rn
   WHERE cond₁ AND cond₂ AND … AND condₖ

Remark: $T(Q) \leq T(R1) \times T(R2) \times \cdots \times T(Rn)$

Key idea: each condition reduces the size of $T(Q)$ by some factor, called selectivity factor
Selectivity Factor

• Each condition $\texttt{cond}$ reduces the size by some factor called selectivity factor

• Assuming independence, multiply the selectivity factors
Example

Q = SELECT * FROM R, S, T WHERE R.B=S.B and S.C=T.C and R.A<40

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 ½

Q: What is the estimated size of the query output T(Q)?
Example

Q = SELECT * FROM R, S, T WHERE R.B=S.B and S.C=T.C and R.A<40

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 ½

Q: What is the estimated size of the query output T(Q) ?

A: T(Q) = 30k * 200k * 10k * 1/3 * 1/10 * ½ = 10^{12}
Selectivity Factors for Conditions

- $A = c$  
  
  $\sigma_{A=c}(R)$  
  
  Selectivity $= 1/V(R,A)$
Selectivity Factors for Conditions

- **A = c** 
  
  /* $\sigma_{A=c}(R)$ */ 
  
  - Selectivity = $1 / V(R,A)$

- **A < c** 
  
  /* $\sigma_{A<c}(R)$ */ 
  
  - Selectivity = ($c - Low(R,A)) / (High(R,A) - Low(R,A))
Selectivity Factors for Conditions

• $A = c$ /* $\sigma_{A=c}(R)$ */
  
  - Selectivity $= 1/V(R,A)$

• $A < c$ /* $\sigma_{A<c}(R)$ */
  
  - Selectivity $= (c - \text{Low}(R, A))/(\text{High}(R,A) - \text{Low}(R,A))$

• $A = B$ /* $R \bowtie_{A=B} S$ */
  
  - Selectivity $= 1 / \max(V(R,A), V(S,B))$
  - (will explain next)
Assumptions

• **Containment of values**: if $V(R,A) \leq V(S,B)$, then all values $R.A$ occur in $S.B$
  
  – 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 $V(S, C)$)
  
  – Note: we don’t need this to estimate the size of the join, but we need it in estimating the next operator
Selectivity of $R \bowtie_{A=B} S$

Assume $V(R, A) \leq V(S, B)$

- A tuple $t$ in $R$ joins with $\frac{T(S)}{V(S,B)}$ tuple(s) in $S$

- Hence $T(R \bowtie_{A=B} S) = \frac{T(R) \cdot T(S)}{V(S,B)}$

$$T(R \bowtie_{A=B} S) = \frac{T(R) \cdot T(S)}{\max(V(R,A), V(S,B))}$$
Size Estimation for Join

Example:

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

(In class…)

\[
= 10K \times 20K / \max(100, 200) \\
= 10K \times 20K / 200 \\
= 1M
\]
Complete Example

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)

• Some statistics
  – T(Supplier) = 1000 records
  – T(Supply) = 10,000 records
  – B(Supplier) = 100 pages
  – B(Supply) = 100 pages
  – V(Supplier, scity) = 20, V(Suppliers, state) = 10
  – V(Supply, pno) = 2,500
  – Both relations are clustered

• M = 11

SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
  and y.pno = 2
  and x.scity = ‘Seattle’
  and x.sstate = ‘WA’
Computing the Cost of a Plan

• Estimate **cardinality** in a bottom-up fashion
  – Cardinality is the **size** of a relation (nb of tuples)
  – Compute size of *all* intermediate relations in plan

• Estimate **cost** by using the estimated cardinalities
Physical Query Plan 1

(On the fly) \( \pi_{\text{sname}} \) Selection and project on-the-fly
-> No additional cost.

(On the fly) \( \sigma_{\text{scity}='Seattle' \land \text{sstate}='WA' \land \text{pno}=2} \)

(Nested loop) \( \text{sno} = \text{sno} \)

Total cost of plan is thus cost of join:
= \( \text{B(Supplier)} + \text{B(Supplier)} \times \text{B(Supply)} \)
= 100 + 100 \times 100
= 10,100 \text{ I/Os}

Supplier (File scan)

Supply (File scan)

\[
\begin{align*}
T(\text{Supplier}) &= 1000 \\
T(\text{Supply}) &= 10,000 \\
B(\text{Supplier}) &= 100 \\
B(\text{Supply}) &= 100 \\
\text{V(Supplier, scity)} &= 20 \\
\text{V(Supplier, state)} &= 10 \\
\text{V(Supply, pno)} &= 2,500 \\
M &= 11
\end{align*}
\]
Physical Query Plan 2

(a) $\sigma_{\text{scity='Seattle' } \land \text{ sstate='WA'}}$

(b) $\sigma_{\text{pno}=2}$

(c) $\pi_{\text{sname}}$

(On the fly)

(Sort-merge join)

(Scan)

Write to T1)

Scan write to T1)

Supplier (File scan)

Supply (File scan)

Total cost

$= 100 + 100 \times \frac{1}{20} \times \frac{1}{10}$ (a)

$+ 100 + 100 \times \frac{1}{2500}$ (b)

$+ 2$ (c) **why?** (output: 1+1)

$+ 0$ (d)

Total cost $\approx 204$ I/Os

M = 11

$T(\text{Supplier}) = 1000$

$T(\text{Supply}) = 10,000$

$B(\text{Supplier}) = 100$

$B(\text{Supply}) = 100$

$V(\text{Supplier, scity}) = 20$

$V(\text{Supplier, state}) = 10$

$V(\text{Supply, pno}) = 2,500$
Plan 2 with Different Numbers

What if we had:
- 10K pages of Supplier
- 10K pages of Supply

(Sort-merge join)

(a) $\sigma_{\text{scity} = \text{'Seattle'} \land \text{sstate} = \text{'WA'}}$

(b) $\sigma_{\text{pno} = 2}$

(c) $\pi_{\text{sname}}$

(d) $\sigma_{\text{sno} = \text{sno}}$

Total cost

$= 10000 + 50$ (a)
$+ 10000 + 4$ (b)
$+ 3 \times 50 + 4$ (c)
$+ 0$ (d)

Total cost $\approx 20,208$ I/Os

Need to do a two-pass sort algorithm

M = 11

V(Supplier,scity) = 20  V(Supplier,state) = 10  V(Supply,pno) = 2,500
Physical Query Plan 3

(On the fly) (d) \[ \pi_{sname} \]

(On the fly)

(c) \[ \sigma_{\text{scity}='Seattle' \land \text{sstate}='WA'} \]

(b) \[ \text{sno} = \text{sno} \]

(Use hash index) 4 tuples

(a) \[ \sigma_{\text{pno}=2} \]

Assume: clustered

(Hash index on pno)

Supply

(Hash index on sno)

Supplier

Total cost

= 1 (a) + 4 (b) \text{ why?} + 0 (c) + 0 (d)

Total cost \approx 5 \text{ I/Os}

T(\text{Supplier}) = 1000 \quad B(\text{Supplier}) = 100 \quad V(\text{Supplier,scity}) = 20 \quad M = 11

T(\text{Supply}) = 10,000 \quad B(\text{Supply}) = 100 \quad V(\text{Supplier,state}) = 10 \quad V(\text{Supply,pno}) = 2,500

B(\text{Supplier}) = 100 \quad V(\text{Supplier,scity}) = 20

B(\text{Supply}) = 100 \quad V(\text{Supply,state}) = 10

V(\text{Supplier,pno}) = 2,500
Histograms

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

Employee(ssn, name, age)

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

σ_{age=48}(Employee) = ?  σ_{age>28 \text{ and } age<35}(Employee) = ?
Employee($ssn$, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee, age}) = 50$

$\min(\text{age}) = 19$, $\max(\text{age}) = 68$

\[ \sigma_{\text{age}=48}(\text{Employee}) = ? \quad \sigma_{\text{age}>28 \text{ and age}<35}(\text{Employee}) = ? \]

Estimate $= 25000 / 50 = 500$

Estimate $= 25000 \times 6 / 50 = 3000$
Histograms

Employee(ssn, name, age)

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

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

<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>
Histograms

Employee(ssn, name, age)

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

σ_{age=48}(Employee) = ?  σ_{age>28 \text{ 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 = 1200

Estimate = 1*80 + 5*500 = 2580
Types of Histograms

• How should we determine the bucket boundaries in a histogram?
Types of Histograms

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

• Eq-Width
• Eq-Depth
• Compressed
• V-Optimal histograms
Employee(ssn, name, age)

Histograms

**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>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

**Eq-depth:**

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..33</th>
<th>33..38</th>
<th>38-43</th>
<th>43-45</th>
<th>45-54</th>
<th>&gt; 54</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 highly frequent values: (48,1900)
V-Optimal Histograms

• Defines bucket boundaries in an optimal way, to minimize the error over all point queries
• Computed rather expensively, using dynamic programming
• Modern databases systems use V-optimal histograms or some variations
Difficult Questions on Histograms

• Small number of buckets
  – Hundreds, or thousands, but not more
  – WHY? *All histograms are kept in main memory during query optimization; plus need fast access*

• Not updated during database update, but recomputed periodically
  – WHY? *Histogram update creates a write conflict; would dramatically slow down transaction throughput*

• Multidimensional histograms rarely used
  – WHY? *Too many possible multidimensional histograms, unclear which ones to choose.*
Multidimensional histograms

Stacked 2D histograms