CS 457: Database Management Systems

Lecture 19
Query Optimization (part 1)
Logistics

• Midterm Exam
  – Won’t return papers
    • You will see some of the same questions again in the final exam (for topics covered by the midterm exam)
    • Review your paper with the TA
  – Curved score = sqrt(original_score) * 10

• Final Exam
  – If your final exam score is higher than midterm, we’ll override your midterm score by the final score
    • Unless you don’t want to…
  – Don’t give up
Logistics

- $F(x) = \sqrt{x} \times 10$
  - 36 -> 60
  - 49 -> 70
  - 64 -> 80
  - 81 -> 90
  - 100 -> 100
Wrap up of Cost Estimation
Know how to compute the cost of a plan

Next: Find a good plan automatically?

This is the role of the query optimizer
Query Optimization Overview

- Parse & Rewrite Query
- Select Logical Plan
- Select Physical Plan
- Query Execution

SQL query

Query optimization

Logical plan

Physical plan

Disk
What We Already Know…

Supplier(sno, sname, scity, sstate)
Part(pno, pname, psize, pcolor)
Supply(sno, pno, price)

For each SQL query….

```
SELECT S.sname
FROM Supplier S, Supply U
WHERE S.scity='Seattle' AND S.sstate='WA'
AND S.sno = U.sno
AND U.pno = 2
```

There exist many logical query plan…
Example Query: Logical Plan 1

\[ \sigma_{\text{sscity} = 'Seattle' \land \text{state} = 'WA' \land \text{pno} = 2} \]

\[ \pi_{\text{sname}} \]

\[ \text{sno} = \text{sno} \]

\[ \text{Supplier} \]

\[ \text{Supply} \]
Example Query: Logical Plan 2

\[ \pi_{\text{sname}} \]

\[ \sigma_{\text{sscity}=\text{Seattle} \land \text{sstate}=\text{WA}} \]
\[ \sigma_{\text{pno}=2} \]

Supplier

Supply
What We Also Know

• For each logical plan...

• There exist many physical plans
Suppliers who supply item number 2 in Seattle, WA:

\begin{equation}
\sigma_{\text{city}='Seattle' \land \text{state}='WA' \land \text{pno}=2} \quad \pi_{\text{name}}
\end{equation}

The physical plan is shown below:

1. **(On the fly)**
   - \( \sigma_{\text{city}='Seattle' \land \text{state}='WA' \land \text{pno}=2} \)
2. **(On the fly)**
   - \( \pi_{\text{name}} \)
3. **(Nested loop)**
   - \( \text{sno} = \text{sno} \)
4. **(On the fly)**
   - Supplier (File scan)
5. **(On the fly)**
   - Supply (File scan)
Example Query: Physical Plan 2

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

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

(On the fly)

(On the fly)

(Index nested loop)

sno = sno

Supplier

(File scan)

Supply

(Index scan)
Query Optimizer Overview

• **Input**: A logical query plan
• **Output**: A good physical query plan
• **Basic query optimization algorithm**
  – Enumerate alternative plans (logical and physical)
  – Compute estimated cost of each plan
    • Compute number of I/Os
    • Optionally take into account other resources
  – Choose plan with lowest cost
  – This is called cost-based optimization
Lessons

• No magic “best” plan: depends on the data

• In order to make the right choice
  – Need to have statistics over the data
  – The B’s, the T’s, the V’s
  – Commonly: histograms over base data
Outline

• Search space

• Algorithm for enumerating query plans
Relational Algebra Equivalences

• **Selections**
  – Commutative: $\sigma_{c_1}(\sigma_{c_2}(R))$ same as $\sigma_{c_2}(\sigma_{c_1}(R))$
  – Cascading: $\sigma_{c_1} \land_c 2 (R)$ same as $\sigma_{c_2}(\sigma_{c_1}(R))$

• **Projections**
  – Cascading

• **Joins**
  – Commutative: $R \bowtie S$ same as $S \bowtie R$
  – Associative: $R \bowtie (S \bowtie T)$ same as $(R \bowtie S) \bowtie T$
Left-Deep Plans, Bushy Plans, and Linear Plans

Linear plan: One input to each join is a relation from disk
Can be either left or right input
Commutativity, Associativity, Distributivity

\[ R \cup S = S \cup R, \quad R \cup (S \cup T) = (R \cup S) \cup T \]

\[ R \bowtie S = S \bowtie R, \quad R \bowtie (S \bowtie T) = (R \bowtie S) \bowtie T \]

\[ R \bowtie (S \cup T) = (R \bowtie S) \cup (R \bowtie T) \]
Laws Involving Selection

\[ \sigma_{C \text{ AND } C'}(R) = \sigma_C(\sigma_{C'}(R)) = \sigma_C(R) \cap \sigma_{C'}(R) \]
\[ \sigma_{C \text{ OR } C'}(R) = \sigma_C(R) \cup \sigma_{C'}(R) \]
\[ \sigma_C(R \bowtie S) = \sigma_C(R) \bowtie S \]

\[ \sigma_C(R - S) = \sigma_C(R) - S \]
\[ \sigma_C(R \cup S) = \sigma_C(R) \cup \sigma_C(S) \]
\[ \sigma_C(R \bowtie S) = \sigma_C(R) \bowtie S \]

Assuming C on attributes of R
Example:
Simple Algebraic Laws

- Example: \( R(A, B, C, D), S(E, F, G) \)
  \[ \sigma_{F=3} \left( R \bowtie_{D=E} S \right) = ? \]
  \[ \sigma_{A=5 \text{ AND } G=9} \left( R \bowtie_{D=E} S \right) = ? \]
Example:
Simple Algebraic Laws

- Example: $R(A, B, C, D), S(E, F, G)$

\[
\sigma_{F=3} (R \bowtie_{D=E} S) = R \bowtie_{D=E} \sigma_{F=3} (S)
\]
\[
\sigma_{A=5 \text{ AND } G=9} (R \bowtie_{D=E} S) = \sigma_{A=5} (R) \bowtie_{D=E} \sigma_{G=9} (S)
\]
Laws Involving Projections

\[ \Pi_M(R \bowtie S) = \Pi_M(\Pi_P(R) \bowtie \Pi_Q(S)) \]

\[ \Pi_M(\Pi_N(R)) = \Pi_M(R) \]
/* note that \( M \subseteq N \) */

- Example \( R(A,B,C,D), S(E, F, G) \)
  \[ \Pi_{A,B,G}(R \bowtie_{D=E} S) = \Pi_? (\Pi_? (R) \bowtie_{D=E} \Pi_? (S)) \]
Laws Involving Projections

\[
\Pi_M(R \bowtie S) = \Pi_M(\Pi_P(R) \bowtie \Pi_Q(S))
\]

/* what are P and Q? see below */

\[
\Pi_M(\Pi_N(R)) = \Pi_M(R)
\]

/* note that M \subseteq N */

- Example R(A, B, C, D), S(E, F, G)
  \[
  \Pi_{A,B,G}(R \bowtie_{D=E} S) = \Pi_{A,B,G} (\Pi_{A,B,D}(R) \bowtie_{D=E} \Pi_{E,G}(S))
  \]
Laws involving grouping and aggregation

\[ \gamma_{A, \text{agg}(D)}(R(A,B) \bowtie_{B=C} S(C,D)) = \]
\[ \gamma_{A, \text{agg}(D)}(R(A,B) \bowtie_{B=C} (\gamma_{C, \text{agg}(D)} S(C,D))) \]
Laws Involving Constraints

Product(pid, pname, price, cid)
Company(cid, cname, city, state)

\[
\Pi_{\text{pid, price}}(\text{Product} \bowtie_{\text{cid} = \text{cid}} \text{Company}) = \Pi_{\text{pid, price}}(\text{Product})
\]
Search Space Challenges

• Search space is huge!
  – Many possible equivalent trees (we just discussed)
  – Many implementations for each operator (previously)
  – Many access paths for each relation (previously)
    • File scan, index

• Cannot consider ALL plans
  – Heuristics: only partial plans with “low” cost
Outline

• Search space

• Algorithm for enumerating query plans