Query Optimization Example

- Sailors (sid, sname, rating, age)
- Boats (bid, bname, color)
- Reserves (sid, bid, day, rname)
- Query:
  
  ```sql
  SELECT S.sid, S.sname, S.age
  FROM Sailors S, Boats B, Reserves R
    B.color = "Red" AND S.age < 30;
  ```

- Reserves has 1000 pages, 10 tuples/page
- Sailors has 500 pages, 50 tuples/page
- Boats has 160 pages, 10 tuples/page
- Data is evenly distributed (assumption)

Steps

- Query optimization steps:
  
  - Translate SQL Query to Relational Algebra Query
  - Create a query tree
  - Create left-deep alternative trees
  - Create query plans for our trees
  - Estimate costs of plans
  - Pick best one
Step 1

- Translate SQL Query to Relational Algebra Query

\[ \pi_{\text{sid}, \text{sname}, \text{age}}(\sigma_{\text{age}<30}(\sigma_{\text{color}="Red"}(\sigma_{\text{bid}=\text{bid}}(B \times \sigma_{\text{sid}=\text{sid}}(S \times R)))))) \]

Step 2

- Create a query tree

\[ \pi_{\text{sid},\text{sname},\text{age}} \]
\[ \sigma_{\text{age}<30} \]
\[ \sigma_{\text{color}="Red"} \]
\[ \sigma_{\text{bid}=\text{bid}} \]
\[ \times \]
\[ \text{Boats} \]
\[ \sigma_{\text{sid}=\text{sid}} \]
\[ \times \]
\[ \text{Reserves} \]
\[ \times \]
\[ \text{Sailors} \]

Note our query only involves joins (as opposed to plain old cross-products) ...

Let's draw the trees with joins
Step 3

- Create left-deep alternative trees
  - We replace cross products if they are really joins
  - How?
    - Using RA equivalences!
  - This is a left-deep plan with joins

\[
\begin{array}{c}
\pi_{sid, name, age} \\
\sigma_{age < 30} \\
\sigma_{color = "Red"} \\
\bowtie_{bid = bid} \\
\bowtie_{sid = sid} \\
\end{array}
\]

Step 3 – More Alternatives

- Would we expect these to have different costs over previous alternative?

\[
\begin{array}{c}
\sigma_{age < 30} \\
\pi_{sid, name, age} \\
\sigma_{color = "Red"} \\
\bowtie_{bid = bid} \\
\bowtie_{sid = sid} \\
\end{array}
\]

No!

\[
\begin{array}{c}
\pi_{sid, name, age} \\
\sigma_{color = "Red"} \\
\sigma_{age < 30} \\
\end{array}
\]

Yes!
Step 4 & 5 – Plans and Estimates

- Let's create some plans and estimates

Total cost: \( 160,160 + 1,002,000 = 1,162,160 \) I/Os !!!

\[ \pi_{\text{sid, sname, age}} (\text{On-the-fly}) \]

\[ \sigma_{\text{color} = \text{"Red"} \land \text{age} < 30} (\text{On-the-fly}) \]

\[ \Join_{\text{sid} = \text{sid}} (\text{Page NL}) \]

\[ \Join_{\text{bid} = \text{bid}} (\text{Page NL}) \]

\[ \text{Boats} \Join \text{Reserves} \]

(On-the-fly)

\[ (\text{this is the cost of reading, not writing}) \]

Step 4 & 5 – Plans and Estimates

- Assume Boats has a clustered B+ Tree on bid (key)
- Assume S has a clustered B+ Tree on sid (key)

Total cost: \( 5,160 + 8,500 = 13,660 \) I/Os !!!

\[ \pi_{\text{sid, sname, age}} (\text{On-the-fly}) \]

\[ \sigma_{\text{color} = \text{"Red"} \land \text{age} < 30} (\text{On-the-fly}) \]

\[ \Join_{\text{sid} = \text{sid}} (\text{Sort Merge}) \]

\[ \Join_{\text{bid} = \text{bid}} (\text{Sort Merge}) \]

\[ \text{Boats} \Join \text{Reserves} \]

*(We are assuming merge (in Sort Merge) takes \( M + N \) ... this may not always be the case. Why?)*
Note on Sort Merge

This is a much better plan than our previous page nested loops one

• But it is an overestimate for Sort-Merge join!
  – In Pass 0 we read all pages then write all pages (2*M)
  – We do not need to read all pages if they are pipelined to the next join (1*M)
  – In last merge pass, we don’t have to write the last set of pages since we always pipeline them to the next operator
  – We also don’t have to read in the file again:
    • Sort(M) + Sort(N) if N needs to be sorted
    • Sort(M) + N if N is sorted

Step 4 & 5 – Plans and Estimates

• Assume Boats has a clustered B+ Tree on bid (key)
• Assume S has a clustered B+ Tree on sid (key)

Total cost: 3,160 + 4,500 = 7,660 I/Os !!!

\[ \pi_{\text{sid}, \text{sname}, \text{age}} \]
\[ \sigma_{\text{color} = \text{"Red"} \land \text{age} < 30} \]
\[ \text{Boats sorted on bid, Reserves isn’t} \]
\[ \text{Assume buffers B = 50} \]
\[ \text{Cost of sorting reserves:} \]
\[ \frac{1 \times 2,000 + 1 \times 1,000}{2} = 1,500 \text{ I/Os} \]
\[ \frac{1 \times 1,000}{2} = 500 \text{ I/Os} \]

\[ \text{Result must be sorted on sid:} \]
\[ \text{Cost of sorting reserves:} \]
\[ \frac{1 \times 2,000 + 1 \times 1,000}{2} = 1,500 \text{ I/Os} \]
\[ \frac{1 \times 1,000}{2} = 500 \text{ I/Os} \]

* We are assuming merge (in Sort Merge) takes M + N … this may not always be the case. Why?
Step 4 & 5 – Plans and Estimates

• Let's try block nested loop join (B=50)

Total cost: $4,160 + 22,500 = 26,660$ I/Os !!!

\[
\begin{align*}
\pi_{\text{sid, sname, age}} & \quad \text{(On-the-fly)} \\
\sigma_{\text{color}=\text{“Red”} \land \text{age}<30} & \quad \text{(On-the-fly)} \\
M + (M/(B-1)) * N = 2,000 + 41*500 = 22,500 & \quad \text{(Block NL)} \\
\bowtie_{\text{sid} = \text{sid}} & \quad \text{(On-the-fly)} \\
M + (M/(B-1)) * N = 160+4*1,000 = 4,160 & \quad \text{(Block NL)} \\
& \quad \text{(this is the cost of reading, not writing)} \\
\bowtie_{\text{bid} = \text{bid}} & \quad \text{(On-the-fly)} \\
\bowtie_{\text{sid} = \text{sid}} & \quad \text{(Block NL)} \\
\text{Sailors} & \quad \text{Boats} \\
\text{Reserves} & \quad \text{Reserves}
\end{align*}
\]

Step 4 & 5 – Plans and Estimates

• Can we do better?

Total cost: $5,410 + 11,500 = 16,910$ I/Os !!!

\[
\begin{align*}
\pi_{\text{sid, sname, age}} & \quad \text{(On-the-fly)} \\
\sigma_{\text{color}=\text{“Red”} \land \text{age}<30} & \quad \text{(On-the-fly)} \\
M + (M/(B-1)) * N = 1,250 + 26*160 = 5,410 & \quad \text{(Block NL)} \\
M + (M/(B-1)) * N = 1,000+21*500 = 11,500 & \quad \text{(Block NL)} \\
& \quad \text{(this is the cost of reading, not writing)} \\
\bowtie_{\text{bid} = \text{bid}} & \quad \text{(On-the-fly)} \\
\bowtie_{\text{sid} = \text{sid}} & \quad \text{(Block NL)} \\
\text{Boats} & \quad \text{Reserves} \\
\text{Sailors} & \quad \text{Sailors}
\end{align*}
\]

How many pages in the answer? 1,250 ... why?
Step 4 & 5 – Plans and Estimates

- Can we do even better?
  - Assume Boats has a hash index on color

Total cost: $56 + 2,050 + 500 + 500 = 3,106$ I/Os !!!!

(less than sort-merge plan)

\[
\pi_{\text{sid, sname, age}} \quad \begin{array}{c}
\bowtie \\
\bowtie \\
\bowtie
\end{array}
\sigma_{\text{age} < 30} \\
\sigma_{\text{color} = \text{“Red”}} \\
\text{Boats}
\]

\[M + \frac{M}{(B-1)}N = 125 + 3\times125 = 500\]

File Scan Sailors = 500
Assume 25% reduction factor

\[M + \frac{M}{(B-1)}N = 50 + 2\times1000 = 2,050\]

Assume 1/16 fewer Reserves match boats
So result is 625 tuples at 5 tuples/page = 125 pages

Boats has 1,600 tuples
Assume 16 colors
Thus, 100 tuples are “Red”
Assume scattered across 50 pages
And the hash index takes 6 pages
Cost = 56

Step 4 & 5 – Plans and Estimates

- Can we still do better?
  - Yes!
  - E.g., we could project on bid after first join ...

- Notice that by adding an index on color to Boats ...
  - We reduced our best query time in half!
  - This is one reason we talk about query optimization
  - It drives physical database design
  - Improving performance (by adding indexes, e.g.) should be driving on how query optimization works!