Outline

- Introduction
- Background
- Distributed DBMS Architecture
- Distributed Database Design
- Distributed Query Processing
  - Query Processing Methodology
  - Distributed Query Optimization
- Transaction Management
- Building Distributed Database Systems (RAID)
- Mobile Database Systems
- Privacy, Trust, and Authentication
- Peer to Peer Systems
Useful References

- Textbook *Principles of Distributed Database Systems*, Chapter 6
Query Processing

high level user query

query processor

low level data manipulation commands
Query Processing Components

- Query language that is used
  - SQL: “intergalactic dataspeak”

- Query execution methodology
  - The steps that one goes through in executing high-level (declarative) user queries.

- Query optimization
  - How do we determine the “best” execution plan?
Selecting Alternatives

\[
\begin{align*}
\text{SELECT} & \quad \text{ENAME} \\
\text{FROM} & \quad \text{EMP, ASG} \\
\text{WHERE} & \quad \text{EMP.ENO} = \text{ASG.ENO} \\
\text{AND} & \quad DUR > 37 \\
\end{align*}
\]

### Strategy 1

\[
\Pi_{\text{ENAME}}(\sigma_{DUR>37 \land \text{EMP.ENO} = \text{ASG.ENO}}(\text{EMP} \times \text{ASG}))
\]

### Strategy 2

\[
\Pi_{\text{ENAME}}(\text{EMP} \bowtie_{\text{ENO}} (\sigma_{DUR>37} (\text{ASG})))
\]

Strategy 2 avoids Cartesian product, so is “better”
What is the Problem?

Site 1
ASG_1 = \sigma_{ENO='E3'}(ASG)

Site 2
ASG_2 = \sigma_{ENO='E3'}(ASG)

Site 3
EMP_1 = \sigma_{ENO='E3'}(EMP)

Site 4
EMP_2 = \sigma_{ENO='E3'}(EMP)

Result

EMP_1\prime = EMP_1 \times_{ENO} ASG_1

EMP_2\prime = EMP_2 \times_{ENO} ASG_2

Site 5
result = EMP_1\prime \cup EMP_2\prime

result_2 = (EMP_1 \cup EMP_2 \times_{ENO} \sigma_{DUR>37}(ASG_1 \cup ASG_1))
Cost of Alternatives

- **Assume:**
  - $size(EMP) = 400$, $size(ASG) = 1000$
  - tuple access cost = 1 unit; tuple transfer cost = 10 units

- **Strategy 1**
  - produce ASG': $(10+10) \times$ tuple access cost
  - transfer ASG' to the sites of EMP: $(10+10) \times$ tuple transfer cost
  - produce EMP': $(10+10) \times$ tuple access cost\(\times2\)
  - transfer EMP' to result site: $(10+10) \times$ tuple transfer cost
  - Total cost

- **Strategy 2**
  - transfer EMP to site 5: $400 \times$ tuple transfer cost
  - transfer ASG to site 5: $1000 \times$ tuple transfer cost
  - produce ASG': $1000 \times$ tuple access cost
  - join EMP and ASG': $400 \times 20 \times$ tuple access cost
  - Total cost

Total cost:

- **Strategy 1:** $460$
- **Strategy 2:** $23,000$
Minimize a cost function

I/O cost + CPU cost + communication cost

These might have different weights in different distributed environments

Wide area networks

- communication cost will dominate (80 – 200 ms)
  - low bandwidth
  - low speed
  - high protocol overhead
- most algorithms ignore all other cost components

Local area networks

- communication cost not that dominant (1 – 5 ms)
- total cost function should be considered

Can also **maximize throughput**
### Complexity of Relational Operations

- **Assume**
  - relations of cardinality $n$
  - sequential scan

<table>
<thead>
<tr>
<th>Operation</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Project</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>(without duplicate elimination)</td>
<td></td>
</tr>
<tr>
<td>Project Group</td>
<td>$O(n\log n)$</td>
</tr>
<tr>
<td>(with duplicate elimination)</td>
<td></td>
</tr>
<tr>
<td>Join</td>
<td>$O(n\log n)$</td>
</tr>
<tr>
<td>Semi-join</td>
<td>$O(n\log n)$</td>
</tr>
<tr>
<td>Division</td>
<td>$O(n\log n)$</td>
</tr>
<tr>
<td>Set Operators</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Cartesian Product</td>
<td>$O(n^2)$</td>
</tr>
</tbody>
</table>
Query Optimization Issues – Types of Optimizers

- **Exhaustive search**
  - cost-based
  - optimal
  - combinatorial complexity in the number of relations

- **Heuristics**
  - not optimal
  - regroup common sub-expressions
  - perform selection, projection first
  - replace a join by a series of semijoins
  - reorder operations to reduce intermediate relation size
  - optimize individual operations
Query Optimization Issues –
Optimization Granularity

- Single query at a time
  - cannot use common intermediate results

- Multiple queries at a time
  - efficient if many similar queries
  - decision space is much larger
Query Optimization Issues – Optimization Timing

- **Static**
  - compilation ⇒ optimize prior to the execution
  - difficult to estimate the size of the intermediate results ⇒ error propagation
  - can amortize over many executions
  - R*

- **Dynamic**
  - run time optimization
  - exact information on the intermediate relation sizes
  - have to reoptimize for multiple executions
  - Distributed INGRES

- **Hybrid**
  - compile using a static algorithm
  - if the error in estimate sizes > threshold, reoptimize at run time
  - MERMAID
Query Optimization Issues –
Statistics

- Relation
  - cardinality
  - size of a tuple
  - fraction of tuples participating in a join with another relation

- Attribute
  - cardinality of domain
  - actual number of distinct values

- Common assumptions
  - independence between different attribute values
  - uniform distribution of attribute values within their domain
Query Optimization Issues – Decision Sites

- **Centralized**
  - single site determines the “best” schedule
  - simple
  - need knowledge about the entire distributed database

- **Distributed**
  - cooperation among sites to determine the schedule
  - need only local information
  - cost of cooperation

- **Hybrid**
  - one site determines the global schedule
  - each site optimizes the local subqueries
Query Optimization Issues – Network Topology

- Wide area networks (WAN) – point-to-point
  - characteristics
    - low bandwidth
    - low speed
    - high protocol overhead
  - communication cost will dominate; ignore all other cost factors
  - global schedule to minimize communication cost
  - local schedules according to centralized query optimization

- Local area networks (LAN)
  - communication cost not that dominant
  - total cost function should be considered
  - broadcasting can be exploited (joins)
  - special algorithms exist for star networks