Query Processing

CISC437/637, Lecture #14

Ben Carterette

Query Evaluation

- To be processed by the DBMS, SQL queries need to be
  1. Parsed and translated into a relational, computable expression
  2. Optimized into an **access path** and **evaluation plan** that minimizes computation/processing time
  3. Evaluated against the actual data

Query Evaluation

- Parsing and translating:
  - Check syntax, verify relations
  - Translate into internal form corresponding to relational algebra expression
- Optimization:
  - Formulate access path tree from relational algebra expression and metadata
  - Formulate evaluation plan
- Evaluation:
  - Execute evaluation plan on indexes + tables to produce result

Goals for this Section

- Produce access paths and evaluation plan
  - Including full algorithmic specification
- Evaluate their cost (mostly in disk I/O)
  - Number of disk seeks * average seek cost
  - Number of pages read * average page read cost
  - Number of pages written * average page write cost

Algorithms for Evaluating Relational Operators

- Many different algorithms, but three common techniques:
  - **Indexing**: if WHERE clause (selection or join), use an index to consider only a small set of records
  - **Iteration**: scan all records in a table in sequence, or all index data entries in sequence
  - **Partitioning**: use sorting or hashing to split input records; replace one expensive operation with a set of cheaper operations on smaller inputs

Selection Algorithms

- We have discussed the cost of selection in the context of indexing
  - Scan cost, equality query cost, range query cost
  - Selectivity, clustered vs unclustered
- That discussion applies to query processing as well
  - Although now we must calculate cost *given* a particular index type, rather than choosing an index type to minimize cost
- Focus on conjunctions and disjunctions

