DB101: Selinger et al. 1979 "Access Path Selection…"

Goetz Graefe – Madison, Wis.

Selinger “Access path selection…” [SIGMOD ‘79]

Agenda

- Context: IBM’s System R
- Logical and physical data model
- Queries (query optimization scope)
- Plan space
- Cardinality and cost estimation
- Exploration and plan selection
- Plan compilation and caching
- Legacy

Research context: IBM’s System R [1974-~1980]

- Codd [1970] Relational data model promised non-procedural data analysis
- IS/1, PRTV – Berkeley Ingres
- Research storage system (RSS), RSI: open-next-close
  - Concurrency control: transactions and queries
  - Write-ahead logging and recovery
  - B-tree indexes, record-to-record pointers (not used)
  - “Sargable” predicates
  - “Scan-driven sort” (no grouping, 4 high keys)
- “Sequel” → “Structured query language SQL”

Logical and physical data model

- Tables stored as “heap files”
- B-tree indexes on multiple columns

No:

- Integrity constraints (unique, not null…)
- Index-organized table (clustered index)
- Hash index, bitmap index, join index
- Views, materialized views

System R queries

- Selection from a single table
- Nested queries in the “where” clause

No:

- “Group by”, “having”
- “outer join”
- “from (select… from… where…) as…”

System R plan space

- Table scan <u>or</u> one index + fetch
- Merge join <u>or</u> (index) nested loops join
- “Interesting orderings”

No:

- Join of two intermediate results (“bushy” plans)
- Index-only retrieval, index intersection, index join
- Cartesian products, semi-join reduction, bit vector filters
- “Poor man’s merge join”, “zigzag” merge join
- Parallel query execution, “poor man’s hash join”

Nested queries

- Ideally using index nested loops join
- Early execution $\Leftarrow$ no correlation

Cardinality estimation

- Row and key counts $\Rightarrow$ selectivity
- “Magic” numbers: 10%, 45%, … 25%, 33%
- Uniformity assumption
- Independence assumption

No:

- Histograms
- Index lookup, b-tree root as quantiles

Cost calculations

- CPU time (# RSI calls) + I/O time (weighted)
- Page counts per segment and index
- Occasional “update statistics”

No:

- Network time
- Memory time
- Robustness (slope of cost function)

Plan space exploration and plan selection

- Dynamic programming (# of tables)
- Interesting orderings (later joins, “group by”, “order by”)
- “the true optimal path is selected in a large majority of cases”
- “Many queries are CPU-bound” $\Leftarrow$ merge join + sort

No:

- Nested queries, sub-query flattening, “query rewrite”
- Grouping, pushing grouping through joins

Plan compilation and caching

- "The minimum-cost [access path] is represented by a structural modification of the parse tree… execution plan in the Access Specification Language (ASL)"
- Compiled to machine code
- Cached as a unit, out-of-date test before run

No:

- Ad-hoc interpretation
- Vectorization
- Parameter-sensitive caching

Legacy of "Access path selection…"

- Dynamic programming, "left-deep" plans: quickly!
- Uniformity assumption, independence assumption
- "Magic" numbers, weighted CPU + I/O
- Compilation and caching: fast applications!
- "Halloween" problem