The Cascades Framework for Query Optimization at Microsoft

Nico Bruno
Cesar Galindo-Legaria
The Cascades Framework

Algebraic representation of logical/physical query fragments

Composable rules (exploration, implementation, enforcers)

Integrated and uniform optimization

Short detour: CascadEGGs?
- Tuned for a specific domain (rule priorities, no cycles, etc.)
- No full propagation of congruences
- Heavy use of properties (derived & required)
- Hierarchical traversal of MEMO for rule binding and application
- Cost embedded in search (properties/pruning)
Cascades at Microsoft

1998
- SQL Server

2008
- Cosmos/Scope

2010
- Parallel Data Warehouse

2014+
- Polybase, SQL On Demand, ...

Sql Server (2001) Scope (VLDBJ’12), PDW (Sigmod’12), Polybase (Sigmod’13)
Rules & Properties

- Execution strategies for SQL subqueries
- Orthogonal optimization of subqueries and aggregation
Statistics

Taxonomy
- Single-column ‘MaxDiff’ histograms
- Multi-column density information
- Average column lengths
- Tries
- HLL / Heavy Hitter sketches (DW / Partitioned tables)
- Skew (Cosmos)

Data sources
- Base tables (including computed columns)
- Filtered indexes
- Materialized views

Create / Update mechanics
- Creation: manual, implicit, automatic
- Update: manual, automatic with mod counts
- Block-level sampling (optional cross-validation)
Cardinality Estimation

Algebra of histograms

• Propagation of statistics through operators
• From arithmetic (WHERE \(a+2 > 5\)) to aggregation (HAVING \(\text{SUM}(a) > 10\))

How do we estimate cardinality values?

• WHERE \(a=10 \text{ AND } b=12 \rightarrow H(a) \& H(b)\)? MCD(a, b)? H(a | b=12)? Depends on skew, correlation, etc.
• QOE (Quality Of Estimation) to rank alternatives
• Rewritten cardinality estimation framework
  • Holistic _calculators_ for estimating query fragments
  • Model assumptions overridable via hints (e.g., ASSUME_JOIN_PREDICATE_DEPENDS_ON_FILTERS)

Other topics

• Autoparameterization and parameter sniffing
• CE feedback / learned cardinalities
• Memory grants are based on cardinality estimation
Costing

**Bottom-up calculation...**

- CPU (e.g., filters) and I/O (e.g., spilling aggs)
- Information: CE, DV, outliers, row sizes, DOP, memory, sorted-ness, etc.
- 3 cost lines: Initial / rewind / rebind

**... with top-down context**

- Row goals
- Bitmap filters
- Estimated rewinds/rebinds
Many design choices in Cascades assume full plan space search!

Plan caches

Trivial plans

Optimization stages and timeouts
- Subset of transformations enabled in each stage
- Best cost at the end of a stage determines next steps

Memo seeding (e.g., Nary-join heuristic reordering)

Various approaches to gradual optimization (e.g., temperature-based)
Supportability

Inputs to QO
- Table, join, query-level hints
- Plan hints: \textit{Find} plan in the Memo!
- Plan guides: Transparently hint queries

Outputs from QO
- Graphical showplan
- Execution traces, including live plans
- DMVs (e.g., \texttt{dm_exec_query_optimizer_info})
Common subexpressions

```
create table D(dkey int not null, dval int)
create unique clustered index idxD on D(dkey)
create table F(fkey int, fval Int, fdkey int not null references D(dkey))
create index idxF on F(fval)

with cte (fval, dval) as (select fval, dval from F join D on fdkey = dkey)
select fval from cte where fval > 5
union all
select dval from cte where dval < 6
```
QO Testing

Dimensions

- Correctness: do we get the same results (modulo non-determinism)?
- Performance: do we get the same performance?
- QO scorecards: other metrics (QO memory, CE errors, plan sizes, etc.)

Data and query sources

- **Massive** test collateral
- Large-scale stochastic testing / Fuzzying
- Benchmarks and performance baselines (microbenchmarks)
- Customer playbacks (e.g., Cosmos Playback)
- MinRepros (Minimizing database repros using language grammars)

Sources of truth

- Different DB system (e.g., Cosmos vs SqlServer)
- Different DB release (e.g., SqlServer’19 vs SqlServer’17)
- Same DB! (Counting, enumerating, and sampling of execution plans in a cost-based QO)

Testing SQL Server’s Query Optimizer: Challenges, Techniques and Experiences, DEBull’08
QO as a Component

AutoAdmin “what-if” index analysis utility
SIGMOD Record’98

Query optimization in Microsoft SQL Server PDW
SIGMOD’12

Figure 1: Overview of the Database Engine Tuning Advisor (DTA).

Figure 2: Overview of query optimization in PDW.
Questions?