

ACM SIGMOD Programming Contest 2018

Finalist Team FloMiGe: Florian Wolf, Michael Brendle, and Georgios Psaropoulos

Task

- Evaluate join queries on a set of pre-defined tables
- Query consists of joins, selections and aggregations
- Preprocessing time of 1 second
- Aggregates of columns to verify correct result

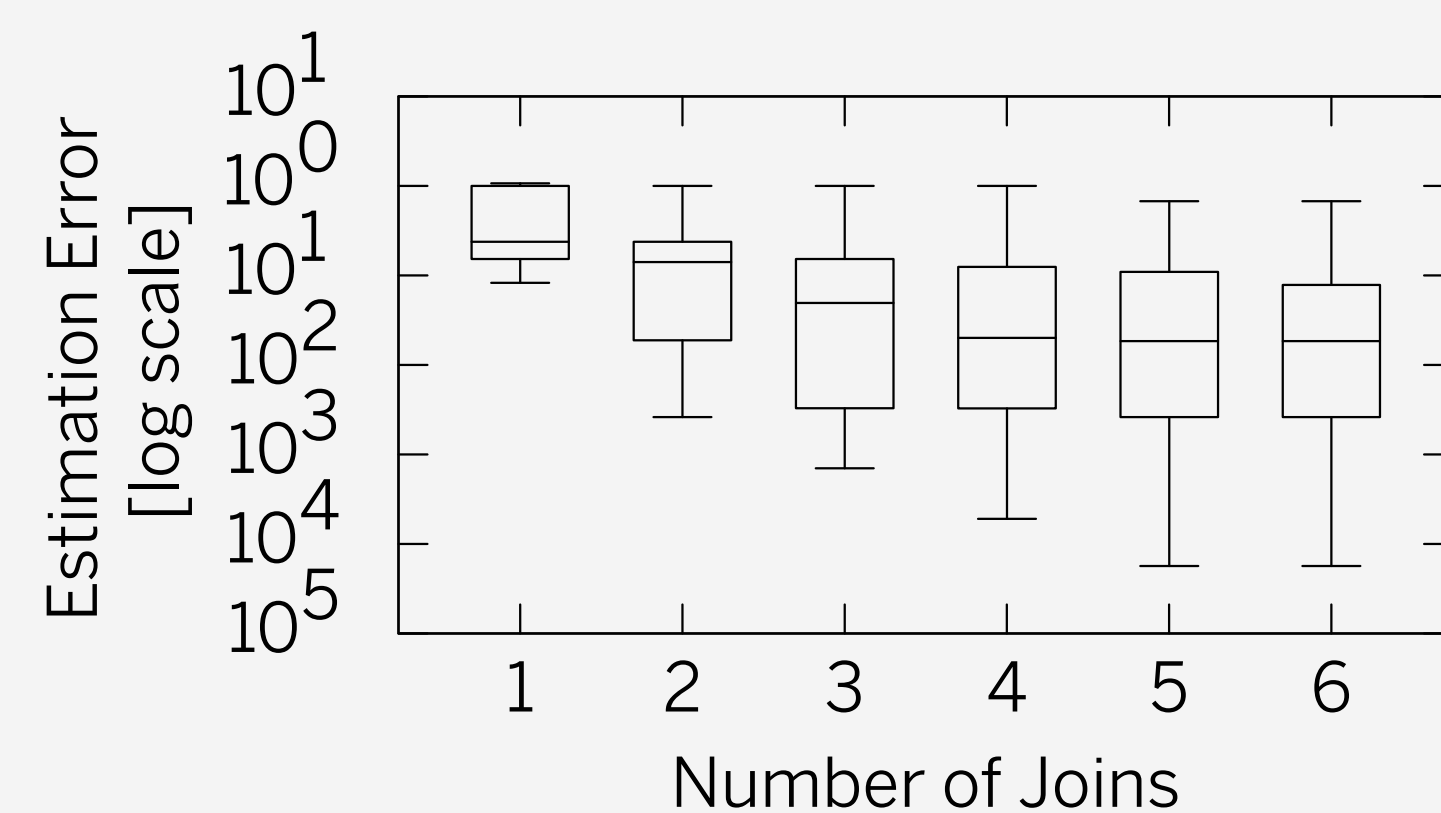
Preprocessing

Determine statistics for each base table column:

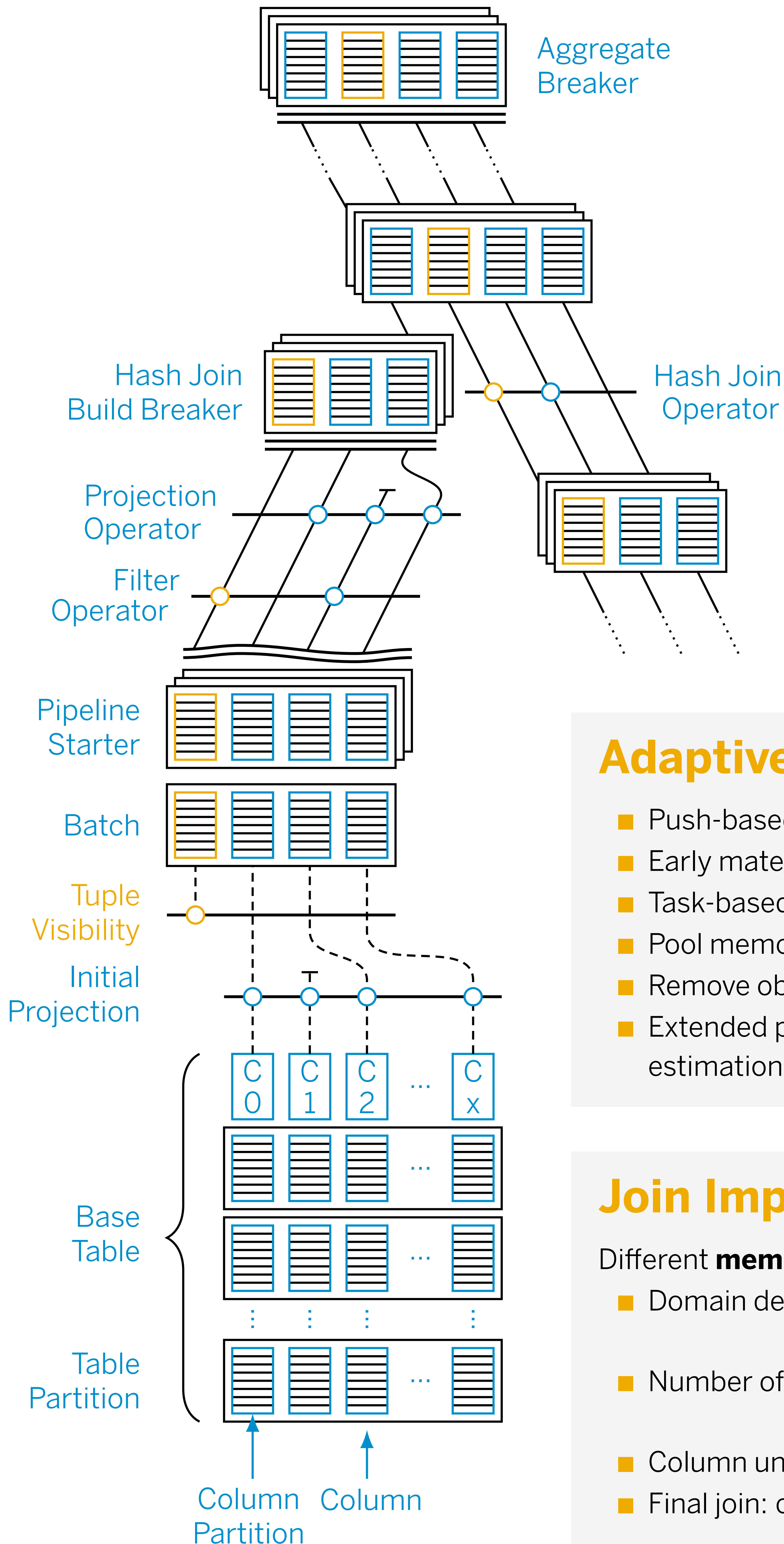
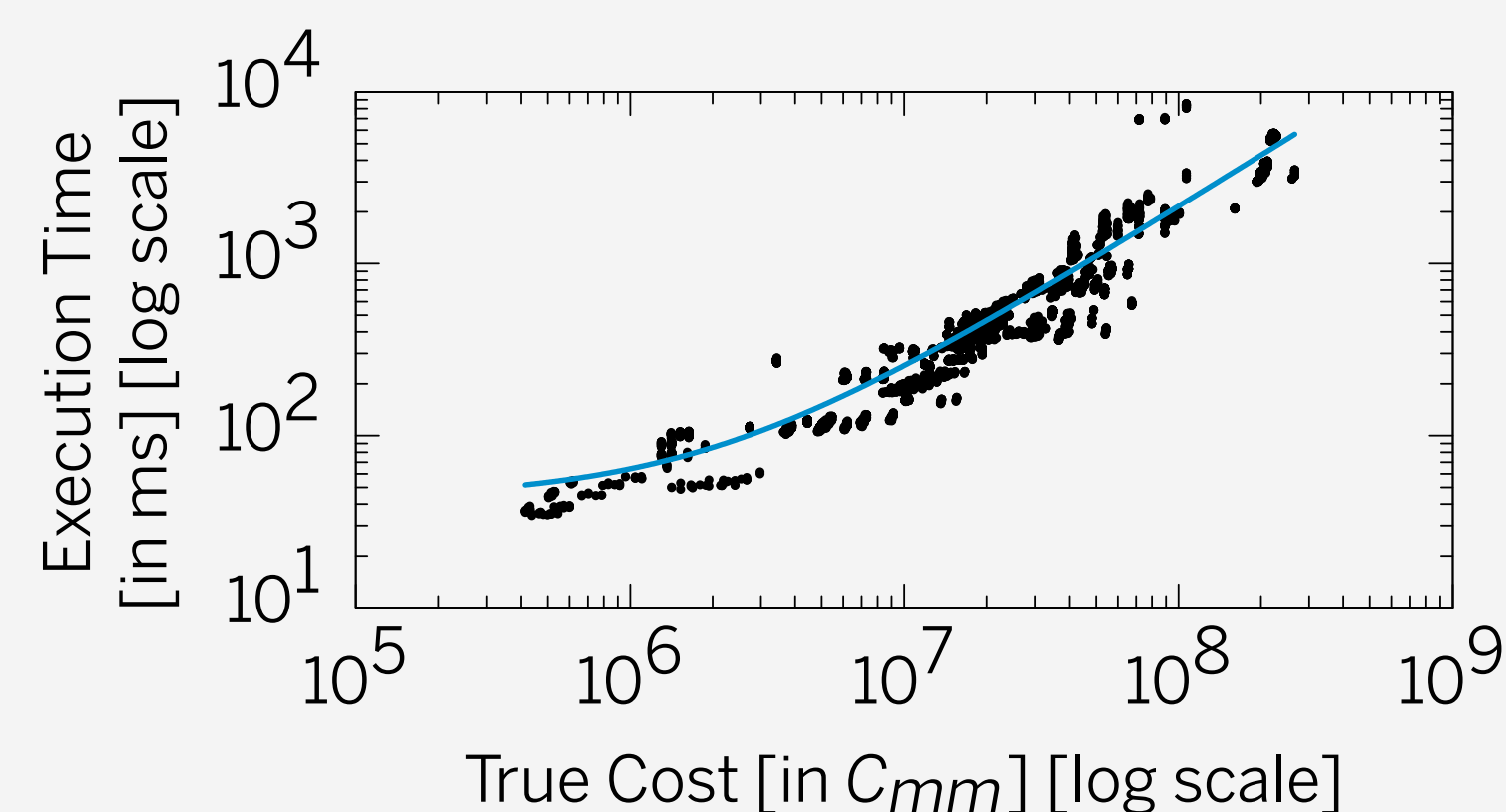
- Min and max value
- Uniqueness

Query Optimizer

- Dynamic programming join enumerator (DP_{size})
- Push down of filter operators
- Simple textbook cardinality estimator



- Main-memory cost function C_{mm}



Adaptive Execution Engine

- Push-based pipelining with batches and bitmap for tuple visibility
- Early materialization
- Task-based parallelism having one task per batch
- Pool memory allocator
- Remove obsolete columns from pipeline through projection
- Extended pipeline breakers to gather runtime statistics, detect estimation errors, trigger re-optimization, and switch to cheaper plan

Join Implementations

Different **memory-efficient hash join** implementations depending on:

- Domain density on build columns: small → Concise Array Table
large → Concise Hash Table
- Number of payload columns: for 1 or 2 payloads → direct storage
otherwise → reference
- Column uniqueness: more efficient build and probe
- Final join: calculate the aggregate during the probe