Cquirrel: Continuous Query Processing over Acyclic Relational Schemas

Qichen Wang[†], Chaoqi Zhang[†], Danish Alsayed[†], Ke Yi[†]
Bin Wu[‡], Feifei Li[‡], Chaoqun Zhan[‡]

Hong Kong University of Science and Technology, Hong Kong, China[†]
Alibaba Group, Hangzhou, China[‡]

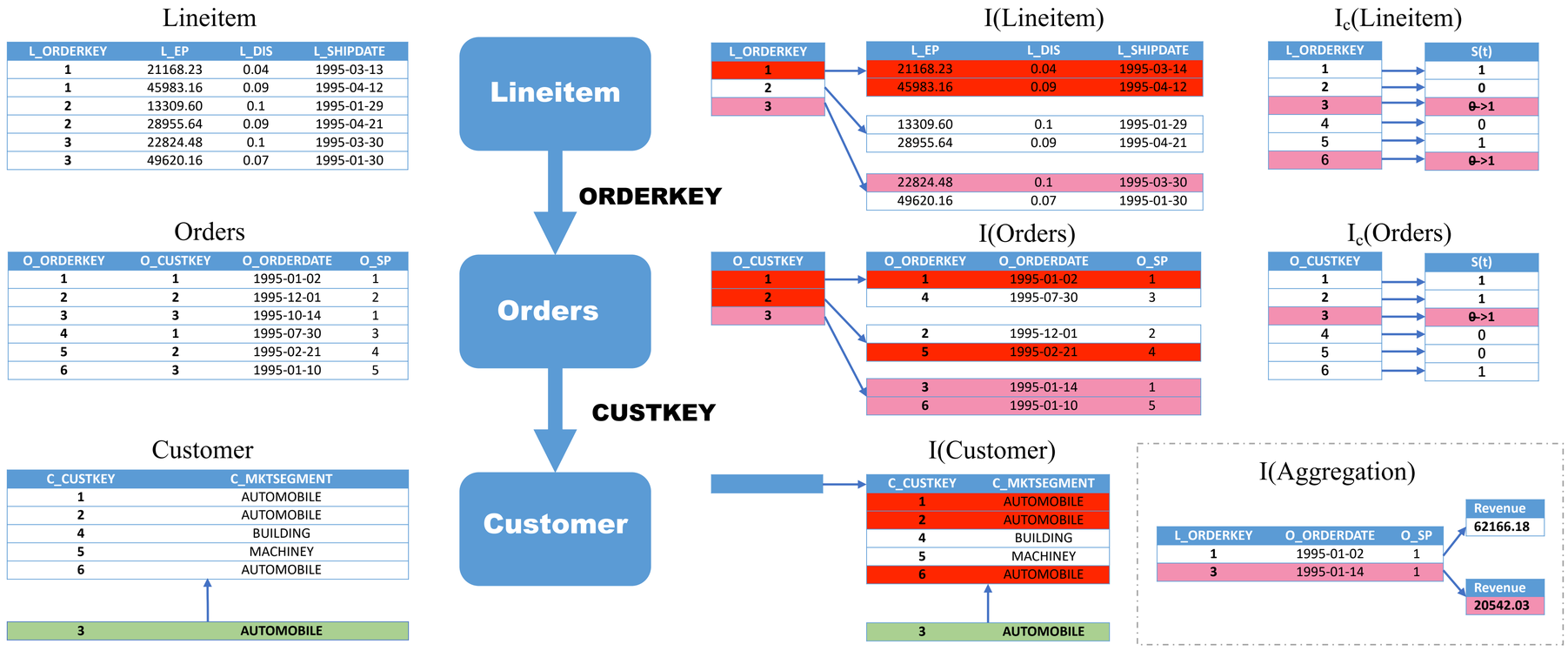

Figure 1: Maintaining TPC-H Query 3 in Cquirrel. Left: Dummy database instance; Middle: Foreign-key Graph; Right: Index for each relation and aggregation. Tuple in red color: alive tuples. Tuple in white color: non-live tuples. Tuple in green color: the insert tuple. Tuple in pink color: the update tuples. All unrelated attributes are omitted from the figure.

1 Motivation & Goal

Motivation: In many emerging applications, queries are evaluated on a database that is being continuously updated. However, existing solutions do not have good support for multi-way join operators.

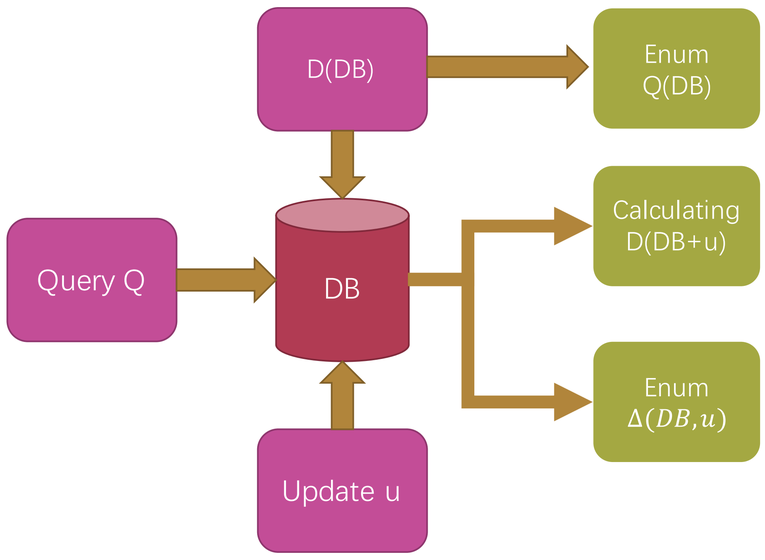

Figure 2: Problem definition for continuous query processing

Goal: In this paper, we demonstrate Cquirrel, a continuous query processing engine built on top of Flink. It provides much better support for multi-way joins than the native join operator in Flink. Meanwhile, it offers better performance, scalability, and fault tolerance than other continuous query processing engines.

2 System Architecture

Cquirrel contains the following main components:

- Core: including execution and maintenance logic of Cquirrel, built on top of Flink DataStream API;
- Code generator: generating execution codes for a given SQL query.
- GUI: for query input, result display, and debug information display.

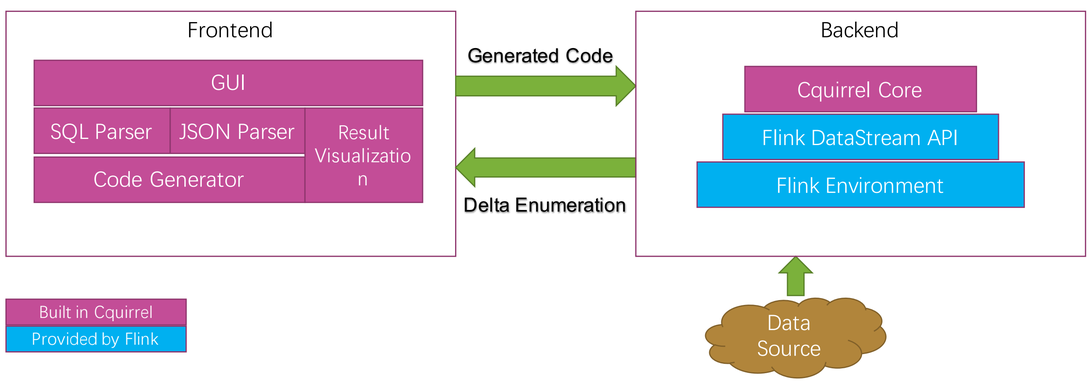

Figure 3: System Architecture

3 Maintenance Procedure

Basic Idea: Maintaining live state of each tuple in the database based on whether the tuple exists in subquery results.

Data Structures: The system needs to maintain an index structure (see Figure 1 right) to keep track of all live tuples. The index should be able to:

1. enumerate all tuples in relation R in constant time per tuple;
2. given any key value v, enumerate all tuples whose value on key x_k is v with a constant delay, or report that there is none;
3. insert or delete a tuple in constant time; and
4. use $O(|R|)$ memory.

4 User Interface

The GUI includes the following components:

- SQL query input;
- Code Generator execution log;
- Execution plan visualization;
- Query result visualization;
- Query result in Table View.

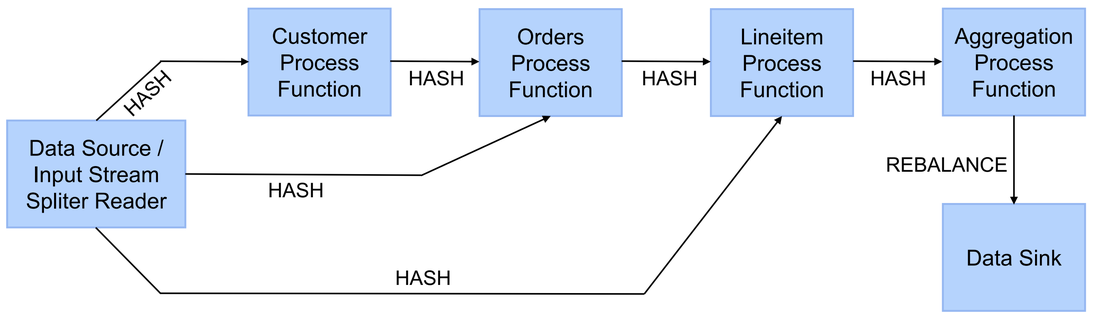

Figure 4: Execution plan visualization

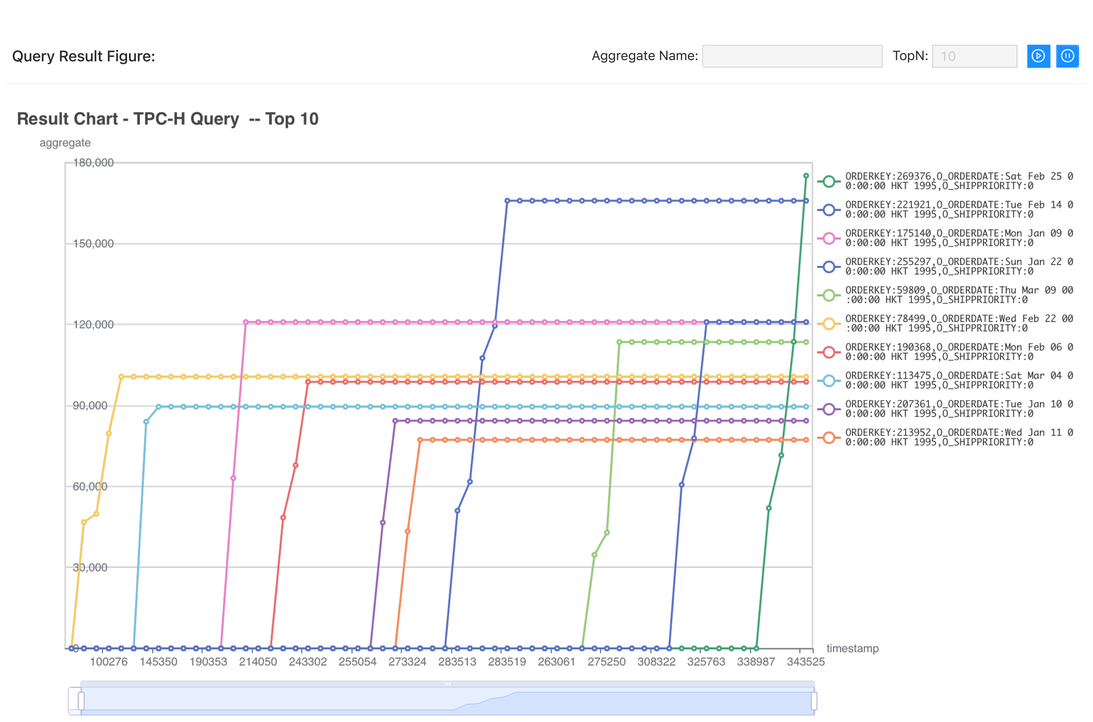

Figure 5: Query result visualization