Exploiting Database Similarity Joins for Metric Spaces

Yasin N. Silva, Spencer S. Pearson
Arizona State University

The Problem

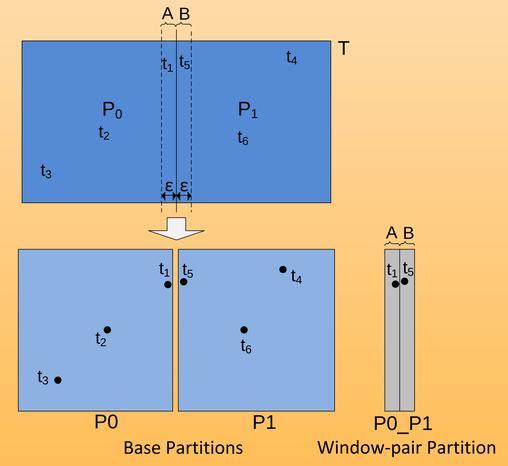
- Similarity joins are a key tool in analyzing and processing data.
- Some standalone Similarity Join algorithms have been proposed.
- Little work on implementing Similarity Joins as physical database operators has been done.

Our Contribution

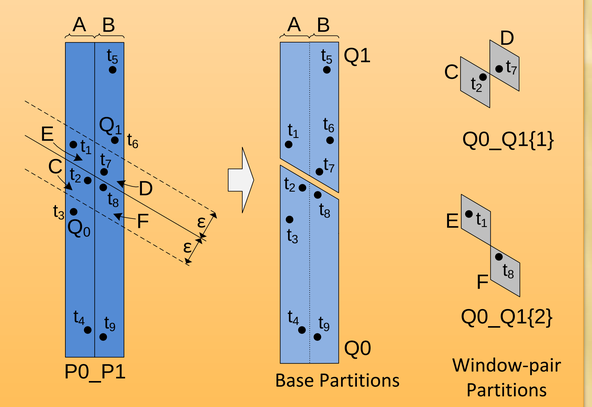
- DBSimJoin, a general Similarity Join database operator for metric spaces implemented inside PostgreSQL.
  - Non-blocking behavior
  - Prioritizes early generation of results
  - Fully supports the iterator interface
- We show how this operator can be used in real-world data analysis scenarios:
  - Identify similar images (vectors)
  - Identify similar publications (strings)

Partitioning in DBSimJoin

- The data is partitioned in a generalized hyperplane using a set of K pivots.
- Two types of partitions exist: base partitions and window-pair partitions.
- Each data record is placed into the base partition of its closest pivot.
- Window partitions hold data that is within ε of the boundary between partitions.



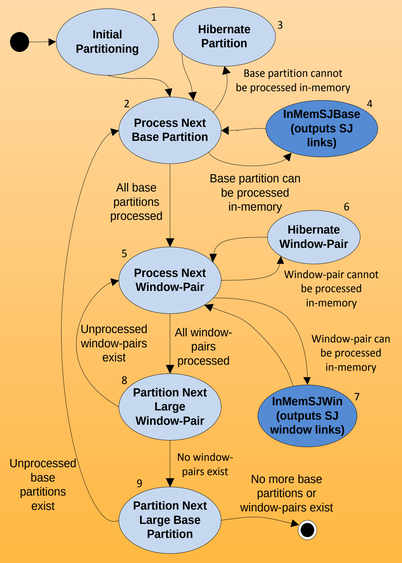
Partitioning a Base Partition



Partitioning a Window Partition

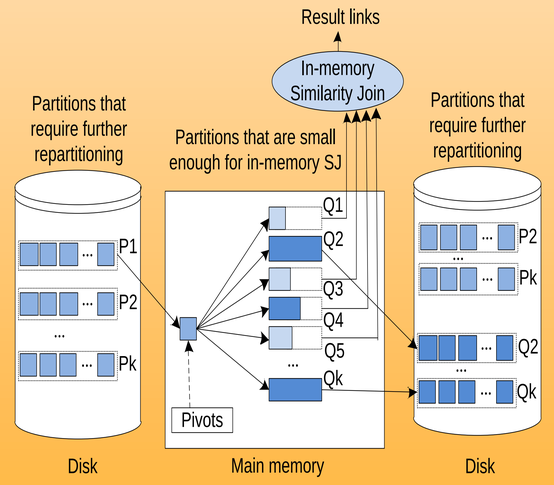
DBSimJoin Algorithm

- Partitions data in successive rounds until the partitions are small enough to be joined with a nested loop.
- Partitioning is done in a series of rounds.
- The algorithm is structured as a finite-state machine in order to support the database iterator interface.



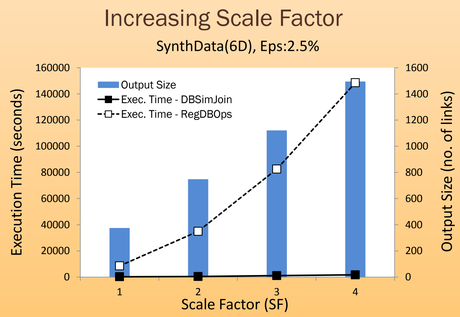
DBSimJoin Rounds

- The first round partitions the input data. All partitions too large to be processed immediately in-memory are stored on-disk.
- Additional rounds re-partition partitions that have been stored on disk.



Performance

Increasing Scale Factor



Increasing Epslion