Exploiting MapReduce-Based Similarity Joins

Yasin N. Silva, Jason M. Reed
New College of Interdisciplinary Arts & Sciences, Arizona State University

Motivation

The Problem

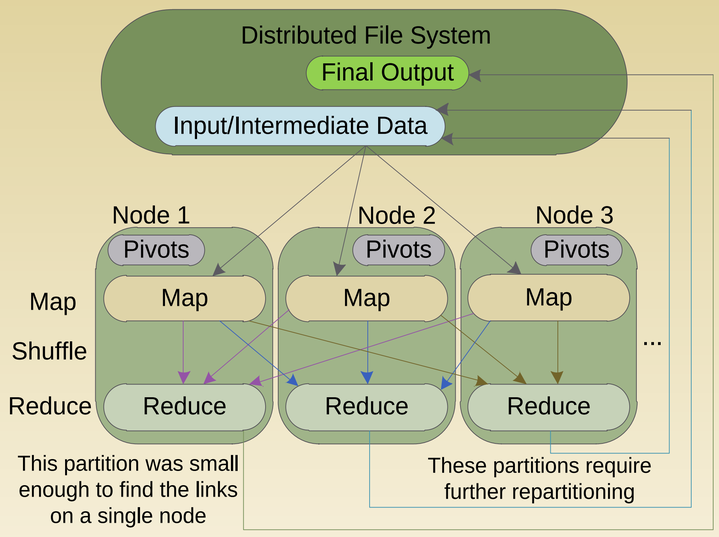
- Cloud-bases systems are crucial to processing and analyzing large data
- Similarity Joins (SJ) are a key data processing and analysis tool
- Very little work on Similarity Joins has been done for cloud systems

Our Contribution

- We propose MRSimJoin –a MapReduce-based algorithm to efficiently solve the SJ problem
- Partitions the data until the subsets are small enough to be processed in a single node
- The algorithm is general enough to be used with data that lies in any metric space
- We have implemented MRSimJoin in Hadoop

MRSimJoin Round

- MRSimJoin iteratively partitions the data into smaller partitions until each partition is small enough to be efficiently processed by a single-node SJ routine
- The process is divided into a sequence of rounds
- The initial round partitions the input data while any subsequent round repartitions a previously generated partition

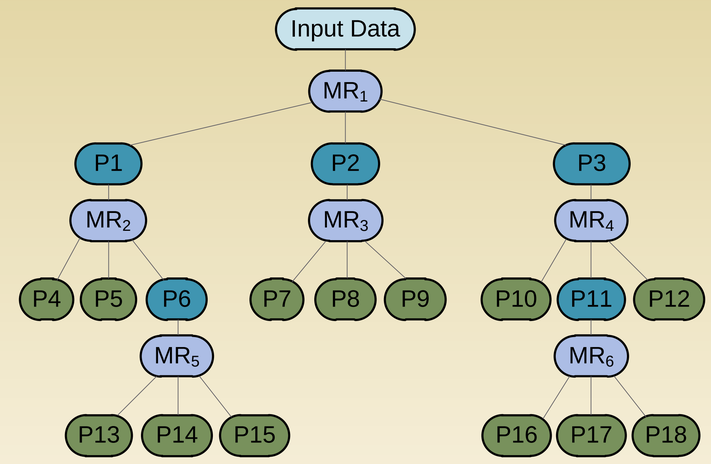


Multiple Rounds

- Each round corresponds to a MapReduce job
- The output of a round includes:
  1) Result links for the small partitions that were processed in a single-node
  2) Intermediate data for partitions that require further partitioning

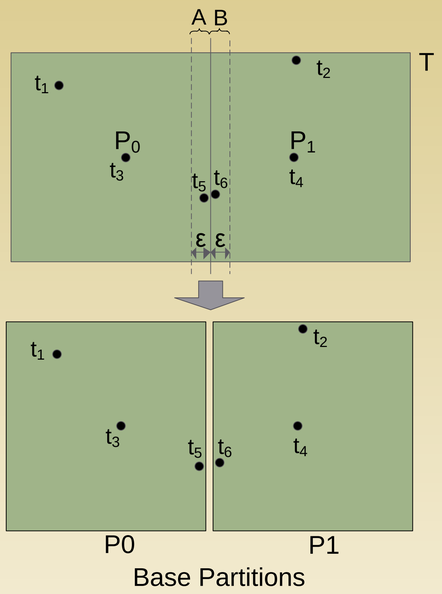
Single-node — The partition is small enough to be solved in a single node. Results written to final output in DFS.

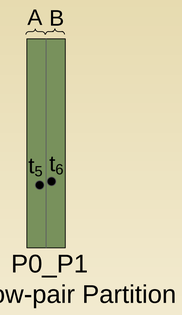
Distributed — The partition will need to be further re-partitioned in additional MapReduce rounds. Intermediate data is written to DFS.



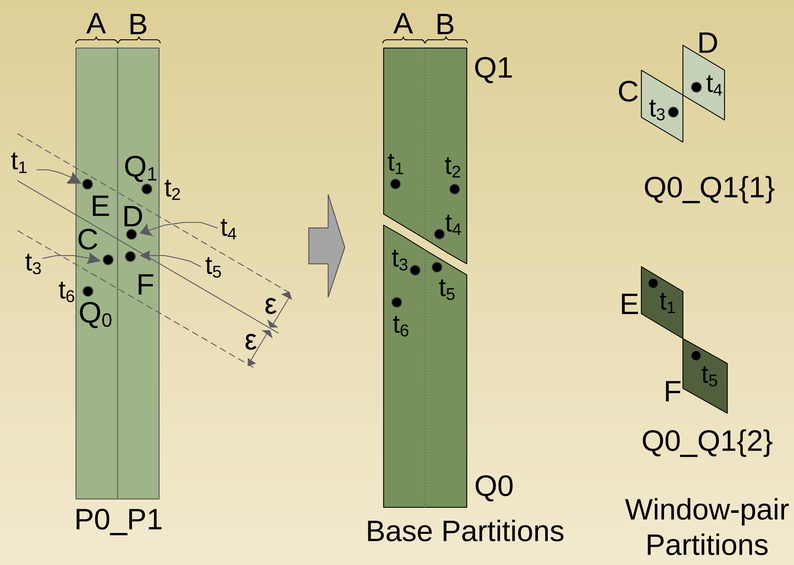
Partitioning in a MRSimJoin Round

- Data partitioning is performed using a set of K pivots (conceptually similar to QuickJoin), which are a subset of the records to be partitioned
- The process generates two types of partitions: base partitions and window-pair partitions
  1) A base partition contains all the records that are closer to a given pivot than to any other pivot
  2) A window-pair partition contains the records in the boundary between two base partitions





Partitioning a base partition



Partitioning a window-pair partition

Performance Evaluation

Tests run over 5 million (SF1) 9D records

Increasing Epsilon - ColorData

Increasing SF - ColorData