Distributed Subtrajectory Join > download here

An open source implementation of the solution proposed in [1].

Joining trajectory datasets is a significant operation in mobility data analytics and the cornerstone of various methods that aim to extract knowledge out of them. In the era of Big Data, the production of mobility data has become massive and, consequently, performing such an operation in a centralized way is not feasible. Here, we address the problem of Distributed Subtrajectory Join (DSJ) processing by utilizing the MapReduce programming model. The problem that we address is as follows: given two sets of trajectories (or a single set and its mirror in the case of self-join), identify all pairs of maximal ``portions'' of trajectories (or else, subtrajectories) that move close in time and space w.r.t. a spatial threshold ε_{sp} and a temporal tolerance ε_{t}, for at least some time duration δt.

Implementation Details

This is a MapReduce solution in Java that has been implemented and tested against Hadoop 2.7.2. The only external library used here is cts for coordinate transformation.

Input Data

The input is an hdfs directory containing csv files (comma delimited) of the form <obj_id, traj_id, t, lon, lat>, where

- obj_id and traj_id are integers (traj_id might be ommited if not available)
- t is an integer corresponding to the unix timestamp
- lon and lat are the coordinates in WGS84

>> MORE INFO