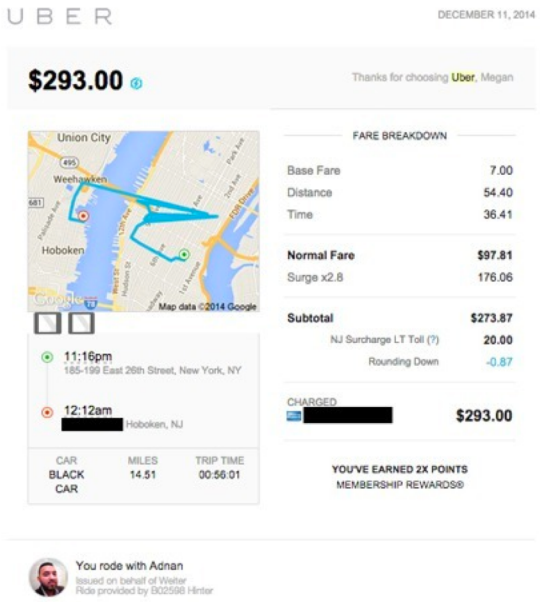
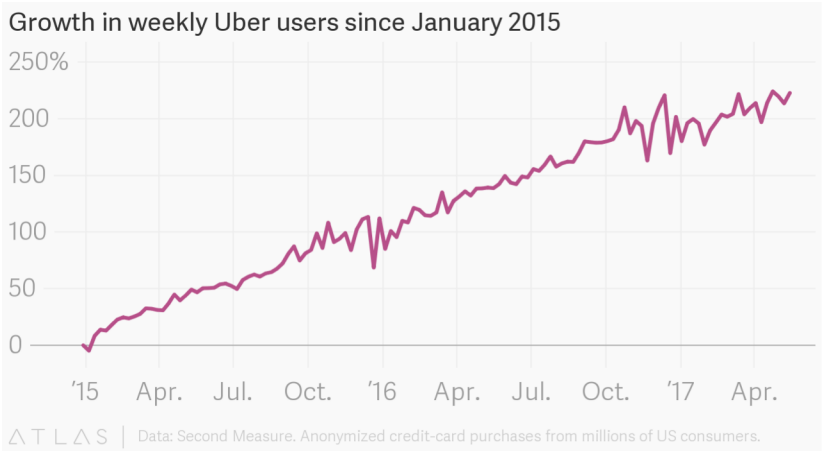
DITA: Distributed In-Memory Trajectory Analytics

Zeyuan Shang(MIT), Guoliang Li(Tsinghua), Zhifeng Bao(RMIT)
zeyuans@mit.edu

Motivation

Trajectory data is getting bigger and bigger

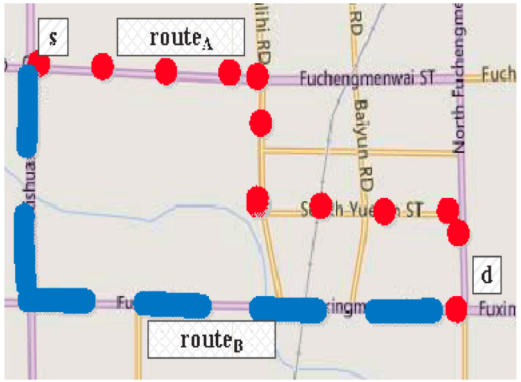




2 Billion Uber trips by 06/2016
62 Million Uber trips in 06/2016

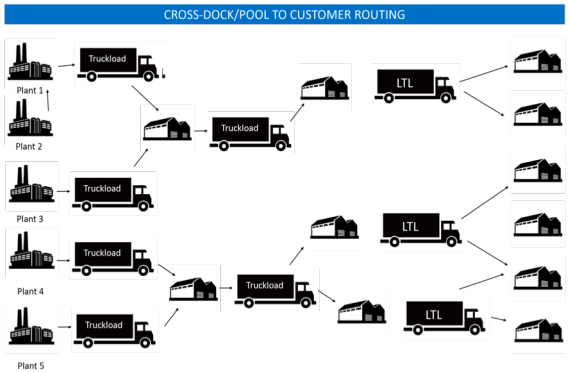
Motivation

Applications of trajectory analytics



Trajectory Recommendation

Road Planning



Transportation Optimization

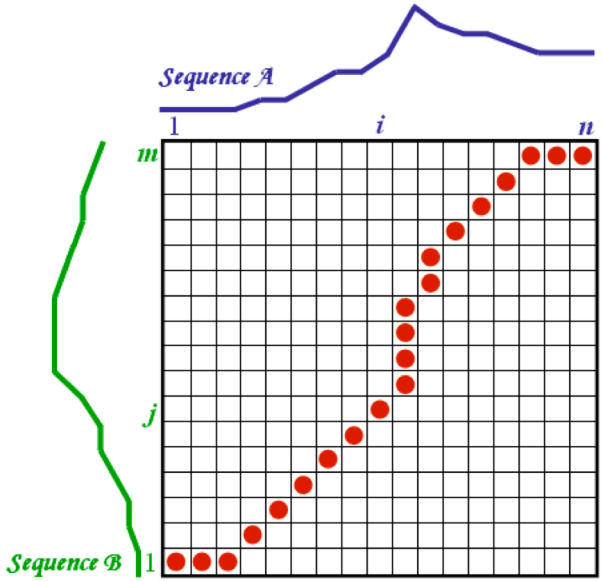
Motivation

Existing systems are limited in a number of ways

- Data locality
- Load balance
- Easy-to-use interface
- Versatility to support various trajectory similarity functions
    - Non-metric ones: DTW, LCSS, EDR
    - Metric ones: Frechet

Background

- Trajectory: a sequence of multi-dimensional points
  - E.g., (1, 2) -> (2, 3) -> (3, 4) -> (5, 5)
- Distance Function between trajectories (e.g., Dynamic Time Warping)



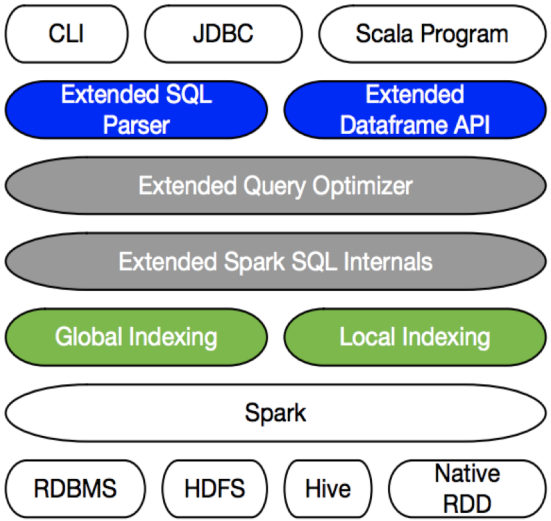
Background

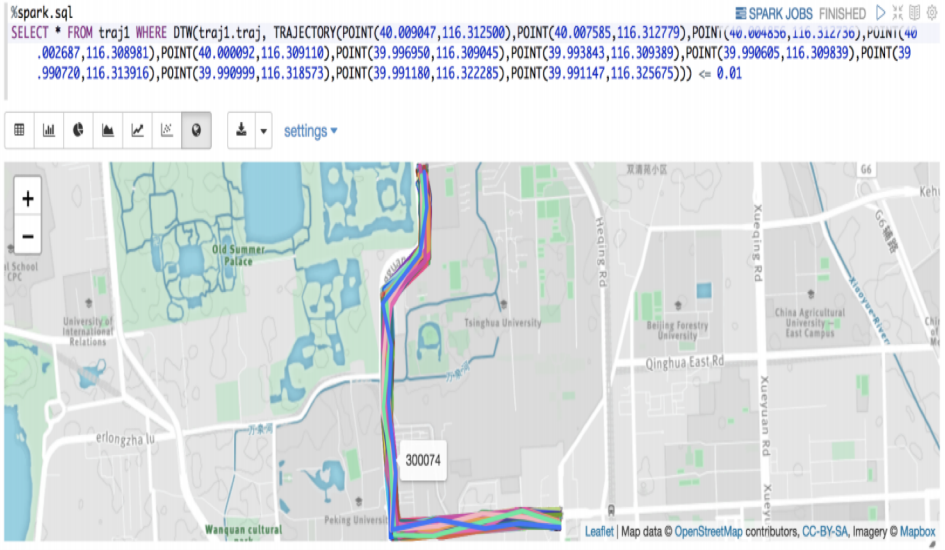
Trajectory Similarity

Given two trajectories **T** and **Q**, a trajectory-based distance function **f** (e.g., DTW), and a threshold τ, if $\mathbf{f(T, Q)} \leqq \tau$, we say that **T** and **Q** are similar.

Overview of System

- Built on Spark SQL
- Support SQL and DataFrame
- Filter-verification framework



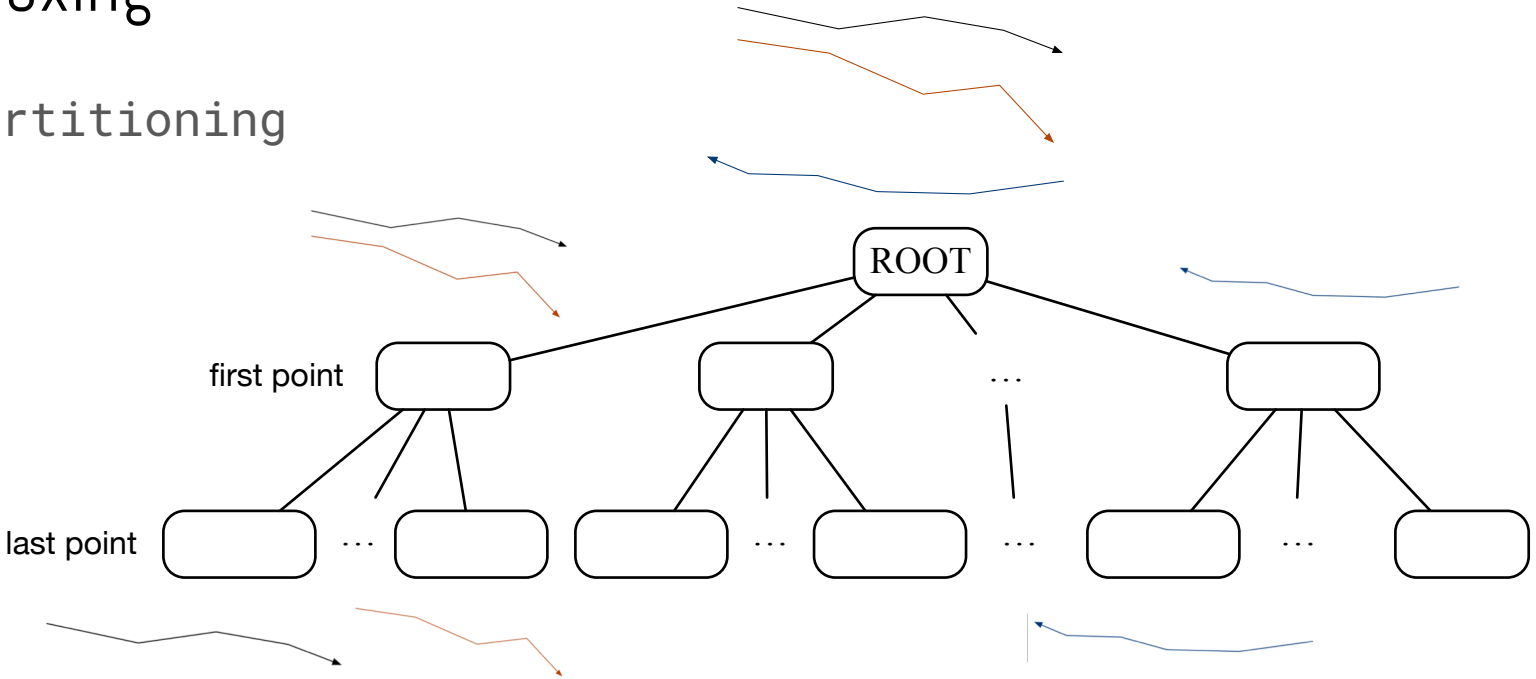


Overview of Methods

- Index
  - Partitioning
  - Global and Local Index
- Trajectory Similarity Search
  - Filter (global + local)
  - Verification
- Trajectory Similarity Join
  - Cost Models
  - Division-based Load Balancing

Indexing

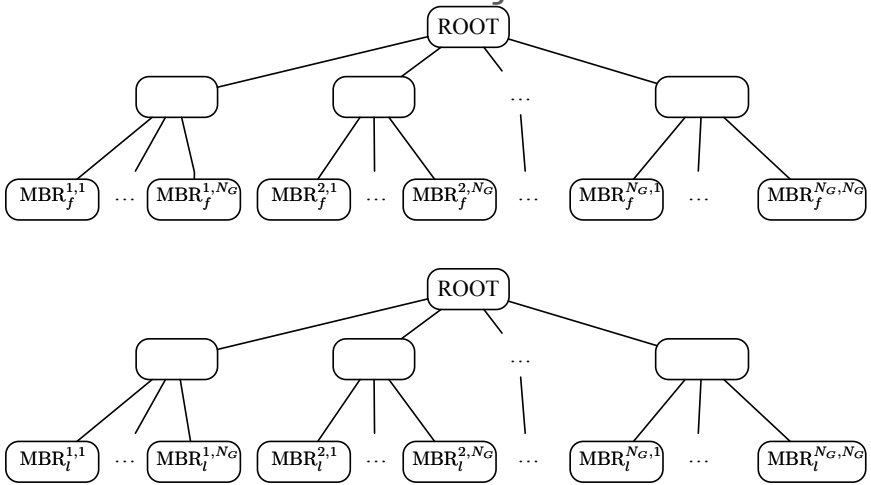
Partitioning

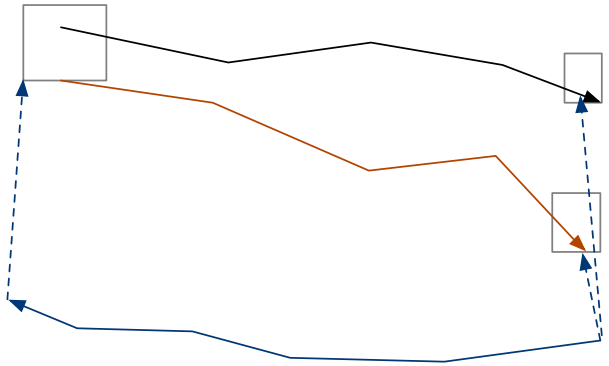


Indexing

Global Index

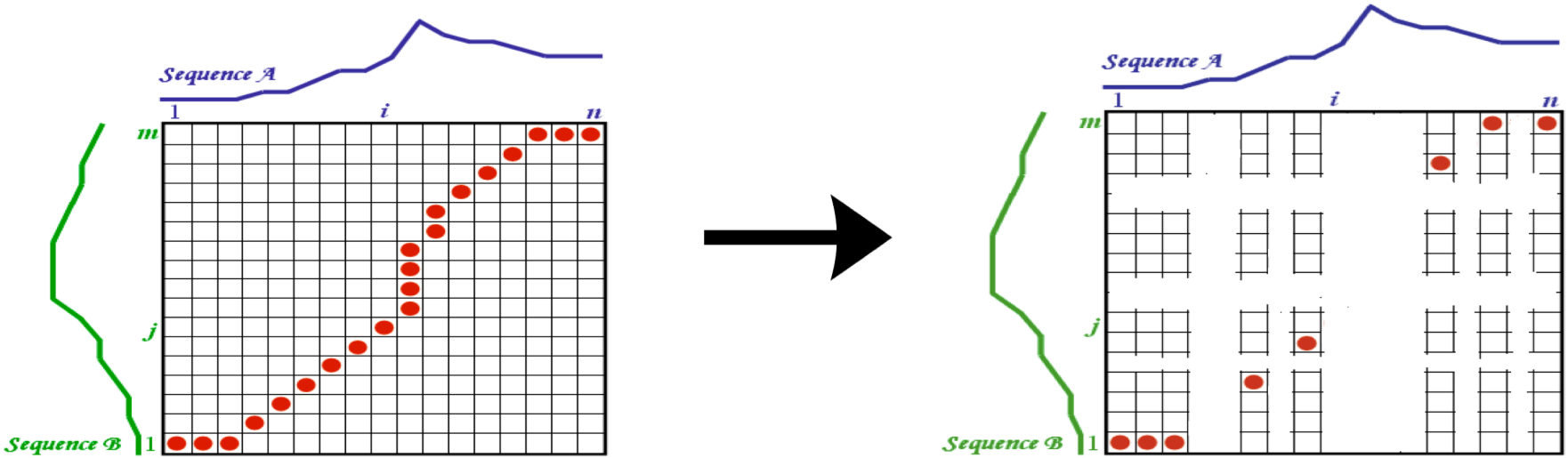
- If MinDist(q, MBR) ≤ **τ**, then for any q ∈ MBR, Dist(p, q) ≤ **τ**
- If MinDist(q, MBR_f) + MinDist(q, MBR_l) > **τ**, then the partition (f, l) doesn't have trajectories similar with q





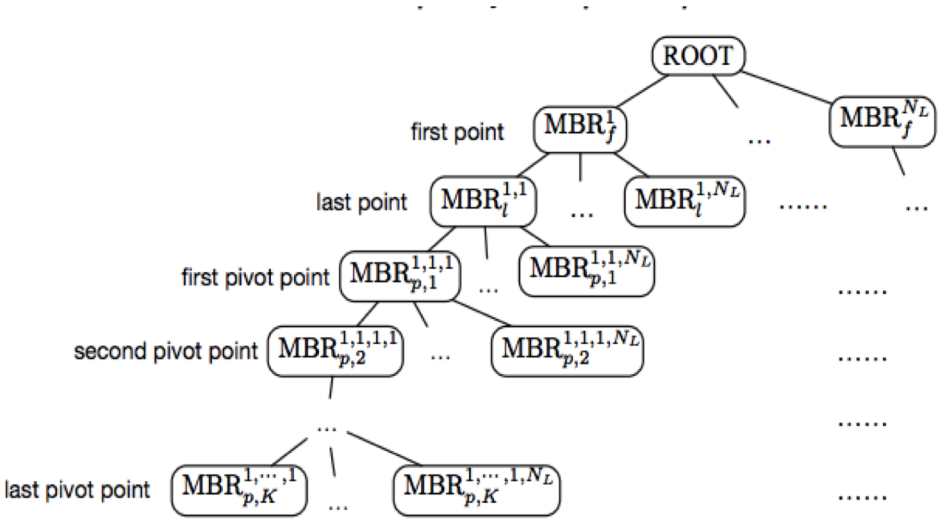
Indexing

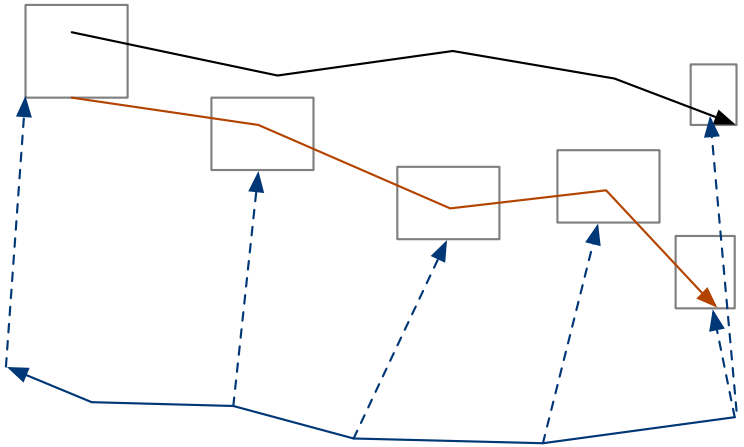
- Pivot Point Based Distance Estimation



Indexing

Local Index



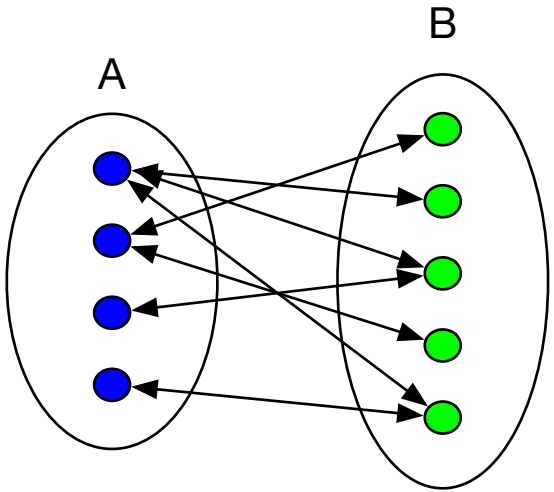


Trajectory Similarity Search

- Basic Idea
  - Global Pruning: find relevant partitions
  - Local Search: find similar trajectories

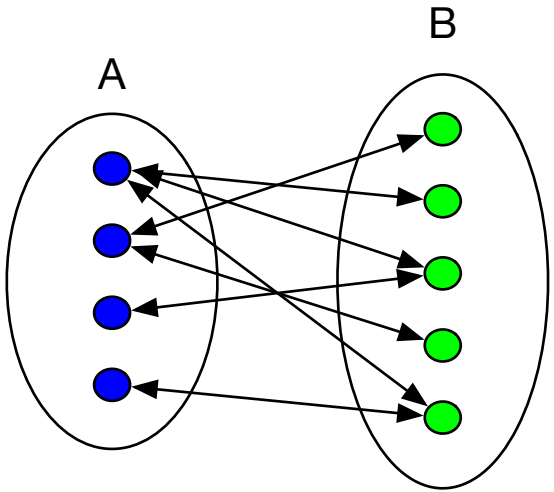
Trajectory Similarity Join

- Cost Models
  - Join Graph
  - Weight of edges (a->b)
    - a sends candidate trajectories to b
    - Transmission cost of a (data transmitted)
    - Computation cost of b (candidate pairs)
  - Built by Sampling



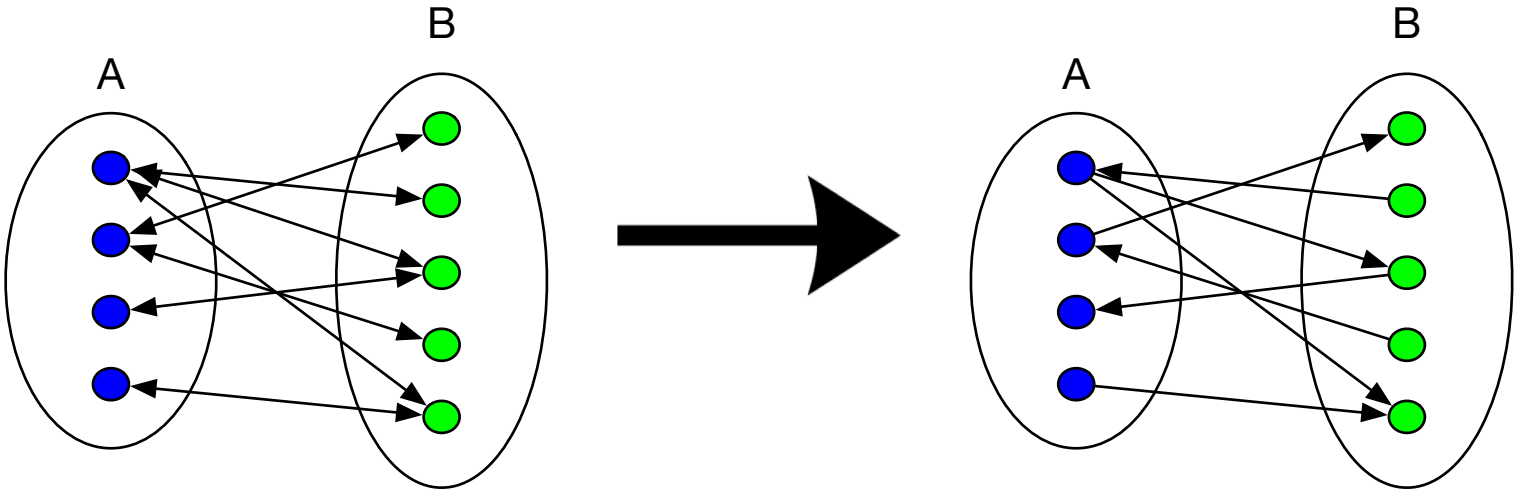
Trajectory Similarity Join

- Cost Models
  - Join Graph
  - Weight of edges (a->b)
    - a sends candidate trajectories to b
    - Transmission cost of a (data transmitted)
    - Computation cost of b (candidate pairs)
  - Built by Sampling
  - Goal: minimize the maximum total cost



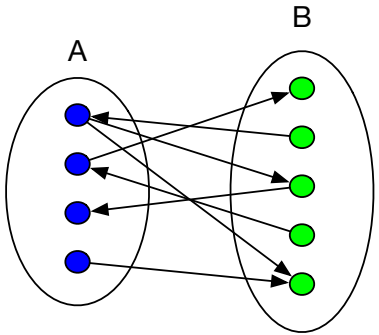
Trajectory Similarity Join

- Graph Orientation

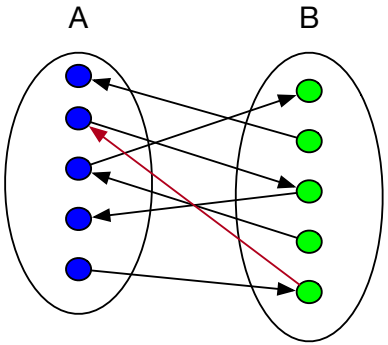


Trajectory Similarity Join

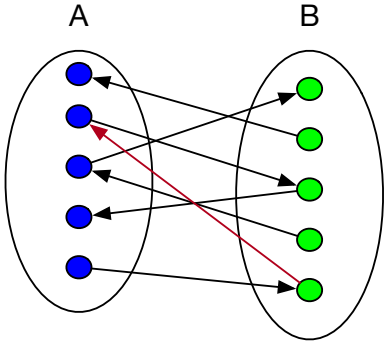
Greedy Algorithm



Initialize



Find partition with largest total cost



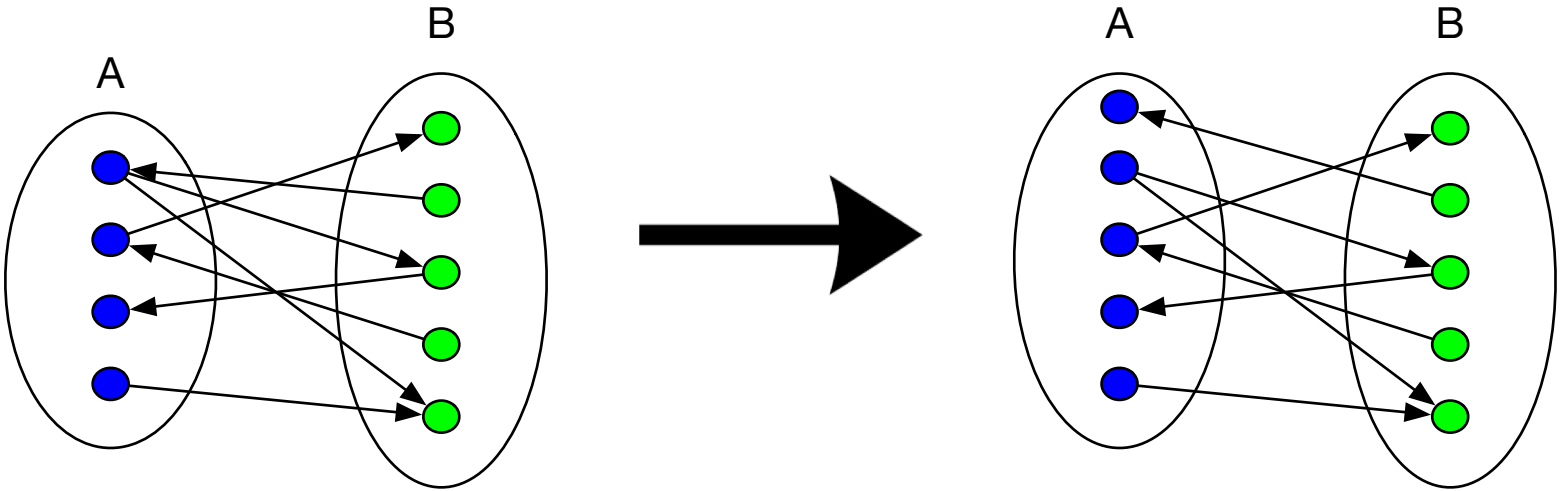
Repeat

Trajectory Similarity Join

- Limitation of Graph Orientation
  - It is greedy
  - Doesn’t work well for partitions with inherently huge cost

Trajectory Similarity Join

- Division-based Load Balancing
  - Division unit: the 98% quantile of total cost
  - For partitions whose total cost bigger than the division unit, we divide them into corresponding number of units



Experimental Results

- Setup
  - 64 nodes with a 8-core Intel Xeon E5-2670 CPU and 24GB RAM
  - Hadoop 2.6.0 and Spark 1.6.0
  - Datasets

Table 2: Datasets

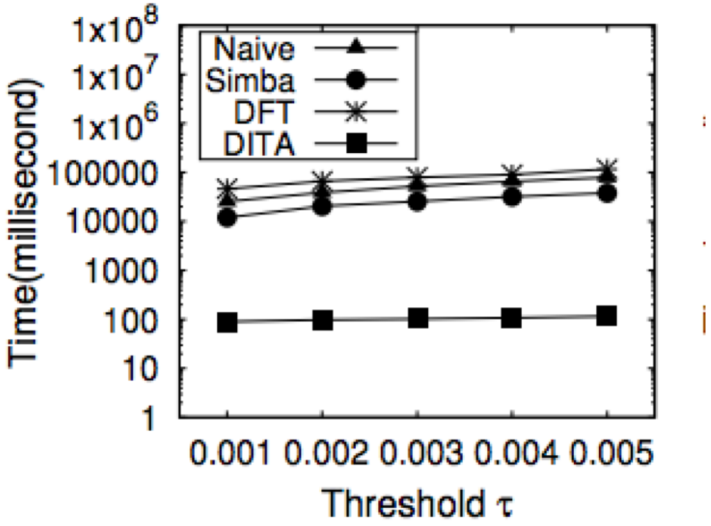
Datasets	Cardinality	AvgLen	MinLen	MaxLen	Size
Beijing	11,114,613	22.2	7	112	10.4GB
Chengdu	15,316,372	37.4	10	209	28GB
OSM(search)	141,236,563	113.9	9	3000	703 GB
OSM(join)	65,764,358	119.5	9	3000	312 GB

Experimental Results

- Baseline Methods
  - Naive
  - Simba (SIGMOD 2016)
  - DFT (VLDB 2017)

Experimental Results

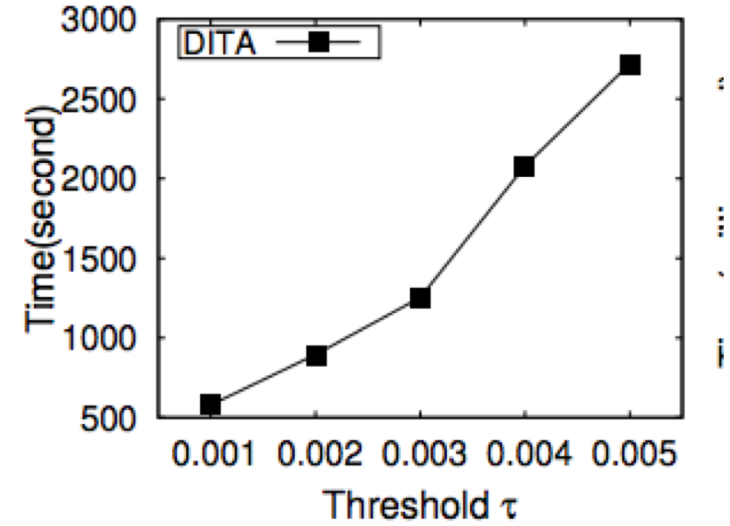
Search on Large Datasets (141M trajectories, 703GB)



(a) Search Time on OSM(search) (DTW)

Experimental Results

Join on Large Datasets (65M trajectories, 312GB)



(b) Join Time on OSM(join) (DTW)

Conclusion

DITA: **D**istributed **I**n-memory **T**rajectory **A**nalytics

- Support trajectory similarity search and join with SQL and DataFrame API
- Support most trajectory distance functions
- Filter-verification Framework
  - Global and Local Index
  - Optimizing Verification
- Experimental results show that DITA outperformed state-of-the-art approaches significantly
- Future Work