

Visualizing Query Processing over large-scale road networks

By:

Paweena Manotipya

Suphanut Jamonnak

Weilong Ren

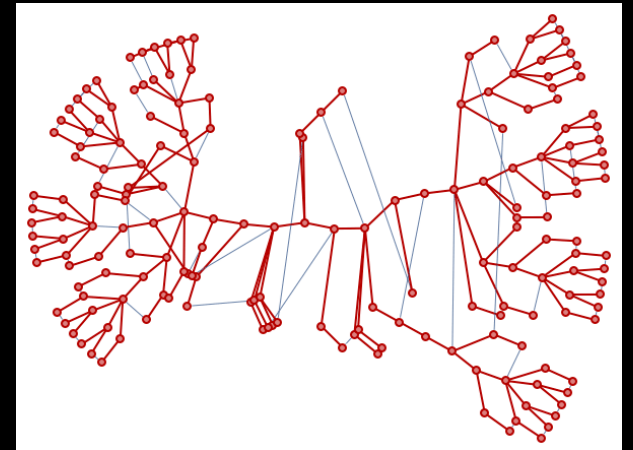
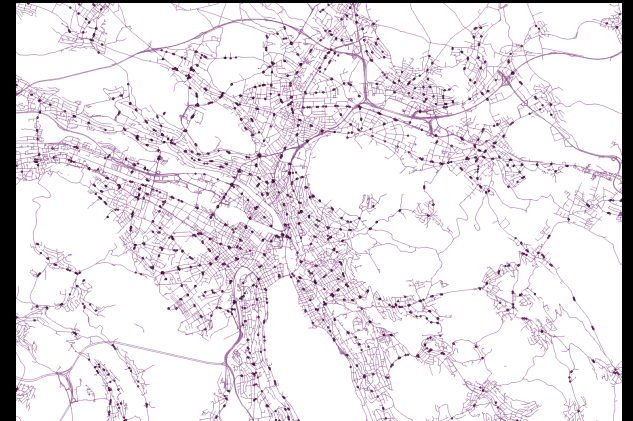
Date: May 01, 2018

Agenda

- What is Road Network?
- Problem Statement
- Data Preparation and Preprocessing
- MapReduce with MongoDB
- System Design and Architecture
- Demonstration
- Conclusion

Road Network

- Road Network: generally forms the most basic level of transport infrastructure within urban areas, and will **link with all other areas**, both within and beyond the boundaries of the urban area.
- Divided into parts such as “intersections”, “urban roads”, “rural roads”, “motorways”, “bicycle lanes”, etc.
- In computer science, Road Network basically illustrated as a **connected graph** and **vertices**.



Problem Statement

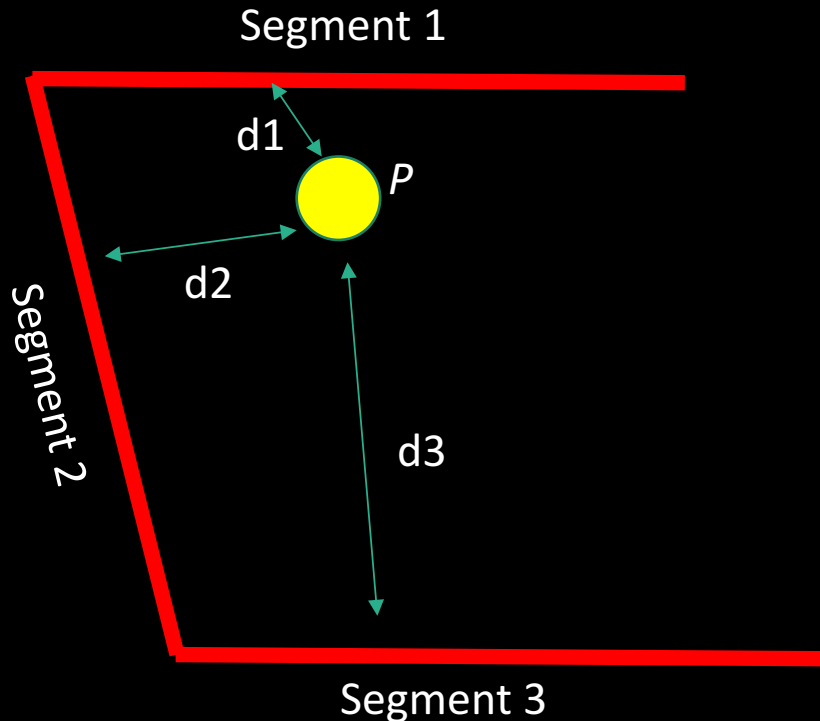
- Existing Applications:
 - Trip Planner: management
 - Gas Station Query: reduce cost
 - Shortest Traveling Time: reduce time
- Problem: These application do not concern about big data.
- For example: Query all Gas Stations in China? Or How to visualize all gas stations in China?
- Our purpose is to find the **efficient** way to **query** and **visualize** a large-scale road networks.

Chicago Crimes

- Develop a visualization tools “Chicago Crimes”.
- Preprocess and Combine 2 datasets including:
- **Chicago Crimes:** reported incidents of crime that occurred in the City of Chicago from 2001 to Present (2018).
 - Extracted in .csv file format (~3GB) with 7 Millions Records. (<https://data.cityofchicago.org>)
- **Chicago Road Network:** Extracted from OSM (open street map API) implemented an extraction tool in python. (~25 MB in .json)



Data Preprocessing



$d1$ is the minimum distance

Add **2** attributes to Chicago Crimes datasets.

P {original, **RoadID**, **LineString**}

- We Store 2 Datasets into MongoDB
- Chicago Crimes (Selected 4 out of 23 attributes)
 - { Case ID, Datetime, Type, Points }
 - Points: 2-D Arrays = [Latitude, Longitude] or Geolocation
 - `Mongo.createIndex('Points', '2dSphere');`
- Road Network (Selected only 2 attributes)
 - { RoadID, LineString }
 - RoadID: Segment ID provided by OSM
 - LineString: Array of Points = [[Lat, Lng], [Lat, Lng], [Lat, Lng], ...]
- Performing “Map Matching” Method

Map Reduce with MongoDB.MapReduce()

- After mapper function we got: { CaseID, Type, Datetime, Points, RoadID, LineString } in our current Chicago Crimes Dataset

- Case1, Road1, 2012, theif
- Case 2, Road1, 2014, murder
- Case 3, Road1, 2014, murder
- Case 4, Road2, 2010, burglary
- Case 5, Road2, 2010, burglary

Mapper Result

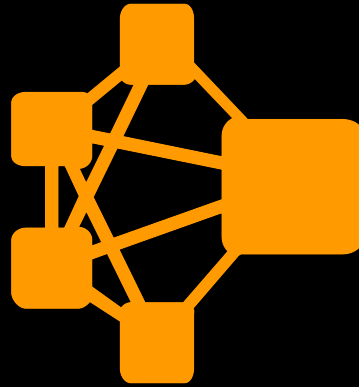
- Key: Road1, Value: [{year: 2012, theif: 1}, {year: 2014, murder: 2} ...]
- Key: Road2, Value : [{year: 2010, burglary: 2} ...]

Reducer Result

System Architectures and Tools



- query



- Mapper
- Reducer



- visualization



Leaflet



Demonstration

Conclusion

- Experience Visualizing and Query Big Dataset using MongoDB and its provided MapReduce framework.
- Performance Improvement by implement our own indexing.
- Combine **trip planner** function:
 - Our system can help user to avoid the dangerous areas.
 - Suggesting safer path during your trip.