Scalability and Adaptability of a Semi-Automated System for Exploring and Fixing OSM Connectivity

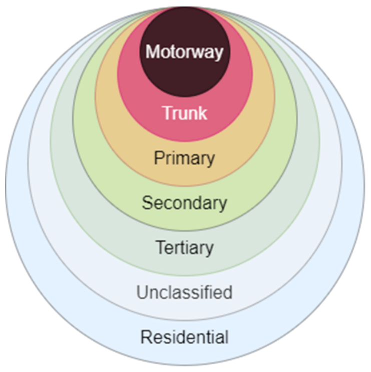
Sarabjot Singh

Professor Mohamed Ali

OVERVIEW & PURPOSE

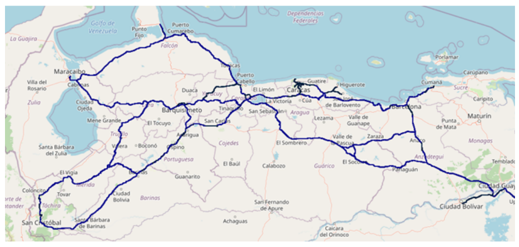
OSM stands for Open Street Maps. It is an open license project with over seven million registered users and 4.5 million map changes each day. Steve Coast created it in 2004 who was inspired by Wikipedia. Open Street Maps is available for all users free of cost. OSM's goal is to produce geographical data services for numerous websites, mobile applications, and hardware devices for free. OSM depends on individual contributors from all over the world, who aim to build a universal map that is free to use and accessible to everyone. Road connectivity needs to be fully achieved for different levels like motorway, trunk, primary, secondary, tertiary, and residential roads to achieve the best results. OSM road network graph has its classifications of geographical entities. The project's open nature makes it prone to missing road networks, road segments, unclassified roads, etc.

The road network is divided into 7 types of road levels. The highest level is Motorway, to the lowest level: residential.

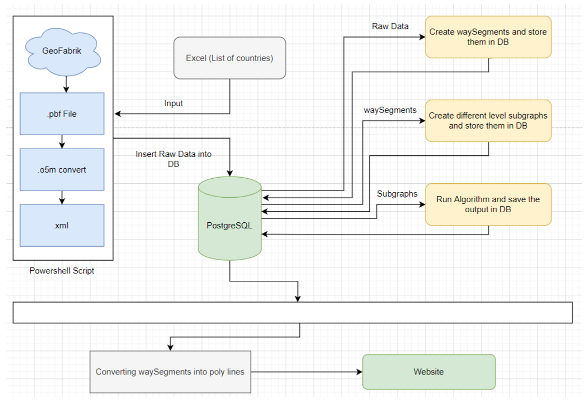


GOALS

To achieve full connectivity for each level of road connection with minimum numbers of lower level roads, and to scale it to multiple regions.



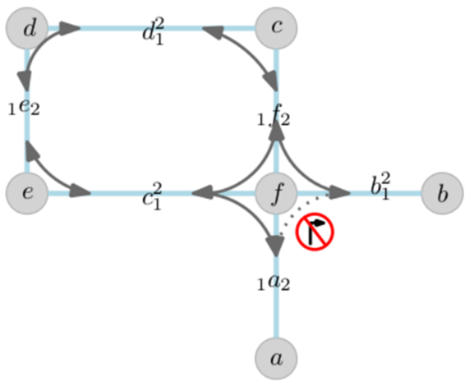
BACKEND ARCHITECTURE


GeoFabrik
.pbf File
.o5m convert
.xml
Powershell Script
Excel (List of countries)
Input
Insert Raw Data into DB
PostgreSQL
Raw Data
Create waySegments and store them in DB
waySegments
Create different level subgraphs and store them in DB
Subgraphs
Run Algorithm and save the output in DB
Converting waySegments into poly lines
Website

- We used powershell script to download, filter, convert and insert geospatial data into the Postgres DB. [Automation]
- We then performed different algorithms on the raw data.
    - BFS
    - Dijkstra shortest path
    - Prim's algorithm
- Created UI interface to show minimum connection needed to achieve full connectivity for each level.

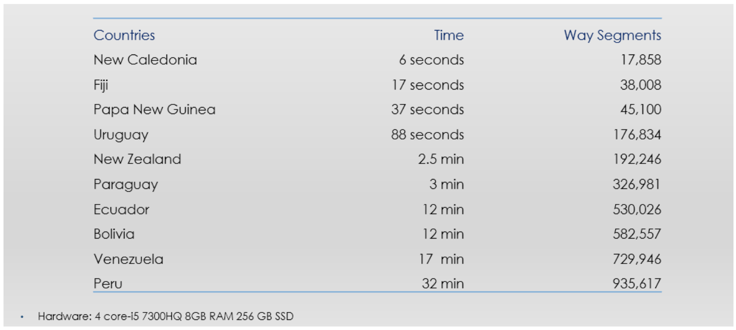
CHALLENGES

1. Handling Turn Restriction
    a. Used custom classes and interfaces



2. Finding shortest path
    a. Implemented Dijkstra's shortest path algorithm using Priority Queue.
3. Removing redundant connections
    a. Implemented Prim's minimum spanning tree to get the shortest path across subgraphs.

RESULTS



Countries	Time	Way Segments
New Caledonia	6 seconds	17,858
Fiji	17 seconds	38,008
Papa New Guinea	37 seconds	45,100
Uruguay	88 seconds	176,834
New Zealand	2.5 min	192,246
Paraguay	3 min	326,981
Ecuador	12 min	530,026
Bolivia	12 min	582,557
Venezuela	17 min	729,946
Peru	32 min	935,617

- Hardware: 4 core-I5 7300HQ 8GB RAM 256 GB SSD

Above is the run-time of the application processing for multiple countries, along with their number of way-segments.

CONCLUSION

We scaled and adapted the current semi-automated system by achieving connectivity on all the map levels and adapting the system to new geographical areas. We harnessed the graph algorithms with some customization to optimize the current process and suggested road connections for each level. A data pipeline was created to help us achieve all our goals.