Short description: The idea is to build database support module for GraphHopper, which provides a high performance imports as well as very large graph analysis on OpenStreetMap using Neo4j-Spatial as a graph database.

Name
Naveen Panwar

Country
India

School and Degree
International Institute of Information Technology, Hyderabad, India.
B.Tech. in Computer Science + MS by research in Spatial Informatics.

Email
panwarnaveen9@gmail.com,
naveen.panwar@research.iiit.ac.in

Phone
+91-9030736015

OSGeo project(s)
Neo4j-Spatial

Title:
To support very large OSM graph analysis integrate Neo4j-Spatial in GraphHopper.

1. Introduction

GraphHopper offers memory efficient algorithms in Java for routing on graphs. E.g. **Dijkstra** and **A*** but also optimized road routing algorithms like: **Contraction Hierarchies**. It directly support OpenStreetMap. Other map data will need a custom import procedure. GraphHopper is written in Java and runs on Linux, OS X, Windows, Raspberry Pi and Android. Other environments which supports at least Java 5 will work too. GraphHopper also works on the Desktop in a Java application without internet access. E.g. you could use the rough user interface called MiniGraphUI provided in the tools module.

2. Background

Neo4j-Spatial is a library of utilities for Neo4j that facilitates the enabling of spatial operations on data. In particular you can add spatial indexes to already located data, and perform spatial operations on the data like searching for data within specified regions or within a specified distance of a point of interest. With the newly released Neo4j-Spatial, any data can be adapted to complex queries with geographic components like "Select all streets in the Municipality of NYC where at least 2 of my friends are walking right now."

In the current implementation of GraphHopper, all data were stored as files. To perform routing or any other operation, the files are read from disc and the data structure is kept in memory or they directly mapped via memory mapping.

Background for Idea -

We can also use standard relational database PostgreSQL(PostGIS) for this purpose, but the problem is RDBMS has its' own known disadvantages like: more memory, synchronization, volume of data or schema for variety of data. Where Neo4J is best suitable choice for this kind of work to us. Also it works fine(similarly faster) for any **Custom Storage**(GraphHopper data).

3. The idea

The idea is to develop a suitable database model using Neo4j-Spatial that fits as an efficient graph storage system for **GraphHopper** and also support the high performance query processing for routing. To store the data in Neo4j-Spatial we have treat every spatial entity as Node and their relationship as Edge in the graph.

Neo4j-Spatial support for GraphHopper will be reached by designing the following modules or class extensions.

- **Configuration Module -**
  Design a module which setup and configure the Neo4j-Spatial libraries with GraphHopper. Also check for the all dependencies of Neo4j and provide an interface to configure the database directory.
- **Export/Import Module -**
  The main functionality of this module is to implement a Custom Graph Storage in Neo4j-Spatial. So the GraphHopper import procedure will able to populate the neo4j database. This module also creates the spatial indexes for these datasets.
- **Query Processing Module -** ***[How to access ?]***
  Implement the interface classes, so that GraphHopper users can easily access the data and

process/query over it while routing on OSM data. Also enable this module to run Dijkstra and Contraction Hierarchy from neo4j

4. Project plan [Detailed Timeline - Thirteen weeks]

Before May 19

- Get familiarise with Neo4j-Spatial and GraphHopper.
- Setup wiki page for the project's weekly reports and discussion.
- Make a repository on github for regular upload and easy access of the source code.

May 19-25

- Write sample function or code in Neo4j, learn/study the mechanism of Neo4j-Spatial and GraphHopper.
- Learn and implementation of various query processing as a sample programming with Neo4j-spatial.

May 26 - June 1

- Write down/ Draw technical design document.
- Learn and understand of the current implementation of data access in GraphHopper.

June 2-8

- Design and implement the Configuration module

June 9-15, June 16-22 [Two weeks]

- Prepare or pre-process the data for storage.
- Build the indexes for Spatial data
- Work on the Import module.

June 23-29

- Mid Term Evaluations
- Testing partitioning functionalities.
- Documentation.

June 30 - July 6, July 7-13 [Two weeks]

- Finish the implementation of export module.
- Check its' import/export module operability.

July 21-27, July 28 - August 3, July 14-20 [Three weeks]

- Design the data access interface in GraphHopper
- Implement classes/interfaces for query processing
- Provide support of Cipher Query Language in GraphHopper

August 4-10

- Testing and bug fixing
- Cleaning up the code
- Wrapping up documentation

August 11-17

- Buffer time
- Testing and bug fixing

August 18-22

- Final evaluations
- Testing and bug fixing (if still something is left over)

After 22nd August

- Work on any issues regarding the particular deployment of project(code).
- Remain an active member of the Neo4j-Spatial and OSGeo community.

5. Future ideas / How can your idea be expanded ?

This idea can be extended to provide applications of Spatial Operation and Spatial Analysis in GraphHopper over the very large OSM data using Neo4j in a very efficient and optimize manner.

Explain how your SoC task would benefit the OSGeo member project and more generally the OSGeo Foundation as a whole ?

If we are enabling GraphHopper for good import in Neo4j, we are opening a whole new possibility of distributed data storage and distributed data processing for spatial graph data in GraphHopper. We also enable the possibility for Neo4j cypher query language to perform spatial queries on OSM data.

Please provide details of general computing experience.

Operating Systems: Linux (Ubuntu, Fedora Xfce), Windows.
Programming Languages: C, C++, Python, Java, Matlab.
Libraries and API: OpenGL, OpenCV, Google Map API (Basic).
Web Development: PHP, HTML, JavaScript, Drupal, AJAX (Basic), Web2py.

Database Management Systems: MySQL, PostGreSQL, SQLite.
Programming Environment: Vim, NetBeans, Eclipse.

Please provide details of previous GIS experience.

I am pursuing my Masters in the field of Spatial Informatics at Lab for Spatial Informatics, IIIT-Hyderabad. I have worked on GIS applications and remote sensing during my graduation. I was also member of team which participated in **ACM SIGSPATIAL GISCUP-2013**. In which our task is to design efficient algorithms about geo-fencing. I am currently working on multi agents system framework for a road traffic system, which will help understand the traffic flows based on a range of interactions between the road network, spatio-temporal patterns of the traffic flow, and driver behavior.

Please provide details of any previous involvement with GIS programming and other software programming.

I was part of VRGeo Project development which is an "Open-Source Collaborative Mapping Platform for Crowdsourcing Location based information" developed by our lab.
I developed android application which provide ***location based service*** for the freelance worker like: plumber, carpenter, electrician etc. to find a suitable job nearby. This app has won first prize for application development at Socinity, IIIT-Hyderabad.

Please tell us why you are interested in GIS and open source software.

Since my freshman year of graduation I am using open source technologies Like: Wikipedia, Linux, Firefox etc. These things always motivated me to learn more and contribute for the development of open-source technologies. and When it comes to operating system I always choose Ubuntu and Fedora Xfce.
During my junior year I get to know about GIS tools like GRASS, QGIS, PostgreSQL etc.. Which gives me an opportunity to work in the very vast field like Data Modeling, Analysis, and Management.

Please tell us why you are interested in working for OSGeo and the software project you have selected.

Neo4j is the only good distributed graph data storage(NoSQL) system and when I worked with it before I found it very interesting. Also they are developing it in direction of spatial data with Neo4j-spatial. Neo4j-spatial allows me to work in both direction - my research area(spatial informatics) as well as a good NoSQL graph database in which the industry is also working on.

So this project will give me an opportunity to get both open source and industry development experience.

Please tell us why you are interested in your specific coding project.
I am always interested in backend programming as I am the Students Web Administrator for my lab. This particular project will allow me to code at low levels of technical design architecture while building scalable storage in Neo4j for GraphHopper. I will get to know about good techniques of optimization and data compression also.

Would your application contribute to your ongoing studies/ degree? If so, how?
As a part of my research work, I am designing Distribute Data Processing and Spatio-Temporal Data Analysis Framework in which I will make use of Neo4j-spatial as it supports OSM data. This project will provide me a very good opportunity to work on this direction.

Please explain how you intend to continue being an active member of your project and/or OSGeo AFTER the summer is over.
I contributed in the development of open-source platform ”VRGeo” , developed in my research center. Since then I am very much passionate to be a part of good open-source programme. GSOC gives me an opportunity to get associated with the OSGeo community and contribute for the benefit of open-source society. I want to pursue my interest with my research work. So I intend to be active member of OSGeo society even after summer.

Do you understand this is a serious commitment, equivalent to a full-time paid summer internship or summer job?
Yes. I am fully prepared for the work and will put in my best efforts. I will be available for 40-50 hours/week work in summer vacation.

Do you have any known time conflicts during the official coding period? (May 19 to August 19).
No, there will be no known time conflicts during the official coding period.