

Graph Databases – Recent development in Neo4j may help accommodate the Geospatial Community

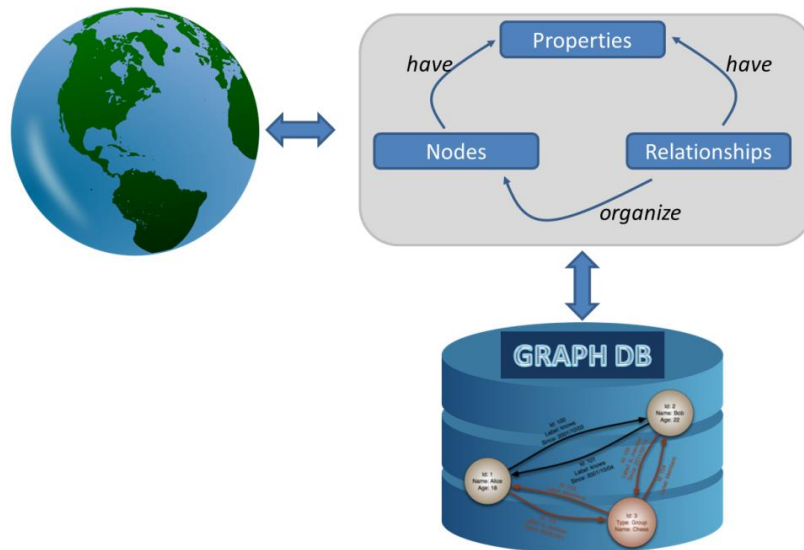
by Emmanuel Stefanakis

Keywords: Graph databases, Neo4j, Neo4j Procedures, NoSQL, Big Geospatial Data, Analytics.

Graph Databases and GIS&T

In the era of big data, graph databases are becoming very popular as they can address important challenges in terms of data size and data complexity. A graph database is a database that uses graph structure with nodes, edges, and properties to represent and store data [1]. In other words, it organizes the content in nodes and relationships instead of tables. Graph databases are especially well suited to applications that involve complex and multiple level relationships between data items, that are well represented as a graph. For this reason, graph databases have been widely used by social networking systems, like Facebook and Twitter, to model and query social relations.

Geospatial-enabled databases and geospatial analytics also involve handling of complex relationships between data items. Road maps or network topologies are typical examples that can be modeled and analyzed using graphs. Hence, a graph database is an obvious direction that Geospatial Information Systems and Technology (GIS&T) should consider.



Neo4j and Neo4j Spatial

In September 2010, a spatial plugin was developed for Neo4j [2], one of the most popular graph database systems. Neo4j Spatial plugin [3] introduced a library of utilities to facilitate the enabling of spatial operations on data. Specifically, Neo4j Spatial includes [4]: (a) utilities for importing geospatial data (e.g., OpenStreetMap data or shapefiles); (b) support for common geometry types (e.g., points, lines, polygons); (c) spatial index mechanisms (R-tree); and (d) limited support for basic spatial and topological operations (such as contain, intersect, touch, and within distance).

Stefanakis, E., 2017. [Graph Databases – Recent development in Neo4j may help accommodate the Geospatial Community](#). *GoGeomatics*. Magazine of GoGeomatics Canada. January 2017.

Six years after, no much development on this direction was made. During this period graph databases and Neo4j Spatial were used in numerous projects; however Neo4j Spatial was not widely adopted by the geospatial community as it was expected. There are several reasons that impeded this to happen [5], which are related to graph databases in general, such as the relational mindset. The transition of database users from the lists and aggregates (as in the relational model) to nodes and relationships has proved a big challenge for graph database systems vendors. In addition, graph databases have been difficult to use due to the lack of a standard language for them. Graph database users must have strong programming skills as they need to write programs and interact with the database through APIs.

Neo4j Procedures and Spatial Procedures

As graph database software becomes more mature and user friendly, it is expected that it will attract more users from the geospatial community. A few months ago, there was a promising development towards this direction. Neo4j version 3.0, released in April 2016, introduced the concept of user defined procedures. A procedure is a mechanism that allows Neo4j to be extended. Procedures are written in java and compiled into jar files. They can be invoked directly from Cypher (the graph query language for Neo4j), take arguments, perform operations, and return results. In other words, procedures can provide implementations of certain functionality that cannot be expressed in Cypher itself.

The geospatial community immediately reacted to this development by releasing a basic set of spatial procedures to support a “much more comprehensive access to spatial from Cypher” [3]. Documentation is not available as yet and the set of procedures is very limited. However, the potential for the expansion of graph databases to GIS&T is huge.

Neo4j Spatial Procedures: A Step closer to GIS&T

I have been offering a technical elective/graduate course on Geographic Databases in the Department of Geodesy and Geomatics Engineering, University of New Brunswick since 2011. Every fall since then, I have discussed with the students the role of NoSQL databases in geospatial applications with special focus on a document stores and graph databases. We have run a lab on MongoDB Spatial and a tutorial on Neo4j Spatial. Using the REST API for creating layers and processing data in Neo4j has always been a barrier for assigning students a hands-on assignment. With the introduction of spatial procedures, we were able last fall to run a lab on Neo4j Spatial as well. Students were offered hands-on experience and used Cypher with spatial procedures to build a simple graph database and perform some basic analytics.

It is anticipated that the set of spatial procedures will grow and become solid in the near future. This will definitely boost the spread of graph databases in the GIS&T and offer a powerful tool for managing and analyzing big geospatial data.

Further Reading

- [1] Graph Database, https://en.wikipedia.org/wiki/Graph_database
- [2] Neo4j, <https://neo4j.com/>
- [3] Neo4j Spatial, <https://github.com/neo4j-contrib/spatial>
- [4] Taverner, C., 2011. Neo4j Spatial: Finding Things Close to Other Things. <https://neo4j.com/blog/neo4j-spatial-part1-finding-things-close-to-other-things/>
- [5] LDDBC – The Graph and RDF Benchmark Reference, <http://www.lddbcouncil.org/>