HydroGraph: Exploring Geographic Data in Graph Databases

Jaudete Daltio[1,2], **Claudia M. Bauzer Medeiros**[2]

[1]Institute of Computing - UNICAMP, Campinas – SP – Brazil

[2]Brazilian Agricultural Research Corporation's - EMBRAPA, Brazil

{jaudete, cmbm}@ic.unicamp.br

Abstract. *Water becomes, every day, more scarce. Reliable information about volume and quality in each watershed is important to management and proper planning of their use. Data-intensive science is being increasingly needed in this context. Associated analysis processes require handling the drainage network that represents a watershed. This paper presents an ongoing work that explores geographic watershed data using graph databases – a scalable and flexible kind of NoSQL databases. The Brazilian Watershed database is used as a case study. The mapping between geographic and graph models is based on the natural network that emerges from the topological relationships among geographic entities.*

1. Introduction and Motivation

During the last decade, the volumes of data that are being stored have increased massively. This has been called the "industrial revolution of data", and directly affected the world of science. Nowadays, the available data volume easily outpaces the speed with which it can be analyzed and understood [Fry 2004]. Computer science has thus become a key element in scientific research.

This phenomenon, known as eScience, is characterized by conducting joint research in computer science and other fields to support the whole research cycle, from collection and mining of data to visual representation and data sharing. It encompasses techniques and technologies for data-intensive science, the new paradigm for scientific exploration [Hey et al. 2009].

Besides the huge volume, the so-called "big data" carries many heterogeneity levels – including provenance, quality, structure and semantics. To try to deal with these requirements, new database models and technologies emerge aiming at scalability, availability and flexibility. The term *NoSQL* was coined to describe a broad class of databases characterized by non-adherence to properties of traditional relational databases [Hecht and Jablonski 2011]. It encompasses different attempts to propose data models to solve a particular data management issue.

Geospatial big data (i.e., big data with a geographic location component) faces even more challenges – it requires specific storage, retrieval, processing and analysis mechanisms [Amirian et al. 2013]. In addition, it demands improved tools to handle knowledge discovery tasks.

The more widely accepted kinds of NoSQL databases include key-value, document, column-family and graph models. Of these, graph databases are the most suitable

choice to handle geospatial big data [Amirian et al. 2014]. Indeed, graphs are the only data structure that natively deals with highly connected data, without extra index structures or joins. No index lookups are needed for traversing data, since every node has links to its neighbors. Besides, in GIS, topological relationships play an important role. These relationships can be naturally modeled with graphs, providing flexibility in traversing geospatial data based on diverse aspects.

Geospatial data about water resources fits these graph connectivity criteria – e.g., watersheds or drainage networks. Owing to the shortage of drinking water, reliable information about volume and quality in each watershed is important for management and proper planning of their use. A watershed is usually represented as drainage network, with confluences, start and end points connected by *drainage stretches* (the network edges).

This paper presents an ongoing work that explores geospatial watershed data taking advantage of graph databases. The goal is to show that this scenario provides additional opportunities for knowledge discovery tasks through classical graph algorithms. The Brazilian Watershed database is used as a case study. The mapping between geospatial and graph models is based on the natural network that emerges from the topological relationships among geographic entities.

The rest of this paper is organized as follows. Section 2 contains a brief description of the main concepts involved and gives an overview of the Brazilian Watershed relational database. Section 3 presents the process of loading watersheds to a graph database and presents results of important and recurrent queries over watersheds. Some research challenges involved are presented in section 4. Finally, section 5 presents conclusions and ongoing work.

2. Research Scenario and Theoretical Foundations

2.1. Brazilian Water Resources Database

Brazil is a privileged country in the water-shortage scenario: it holds 12% of the world total and the largest reserve of fresh water on Earth [Brebbia and Popov 2011]. Its distribution, however, is uneven across the country. Amazonas, for instance, is the state with the largest watershed and one of the less populous in Brazil. Furthermore, some rivers are being contaminated by waste of illegal mining activities (such as mercury), agricultural pesticides, domestic and industrial sewage leak and garbage.

Reliable information about volume and quality in water resources is extremely important to management and proper planning of their use. To this end, the Brazilian Federal Government approved in 1997 the National Water Law [Brazil 1997] aiming to adopt modern principles of management of water resources and created in 2000 the National Water Agency (ANA), legally responsible for accomplishing this goal and ensuring the sustainable use of fresh water.

To organize the required data and support management tasks, ANA adopts the watershed classification proposed by Otto Pfafstetter [Pfafstetter 1989], constructing a database that covers the entire country, named *Brazilian Ottocoded Watershed*. This database represents the hydrography as a drainage network: a set of drainage points and stretches. This network is represented as a binary tree-graph, connected and acyclic,

whose edges – the drainage stretches – go from the leaves to the root, i.e., upstream to downstream.

The Brazilian drainage network is composed by 620.280 drainage points (vertices, in graph terms) and 620.279 drainage stretches (edges). Drainage points represent diverse geographic entities:

(i) a watercourse start point, usually a spring or water source;
(ii) a watercourse end point, usually a river mouth;
(iii) a stream mouth point, which flows into the sea; and
(iv) the shoreline start or end point, two reference points in the coast (one of each) that delimit the shoreline line, being the integrating elements of the entire drainage system.

The first three kinds of drainage points can be seen in Figure 1. The degree of a drainage point represents its valence, value 1 represents start or end points and value 3 represents confluences.

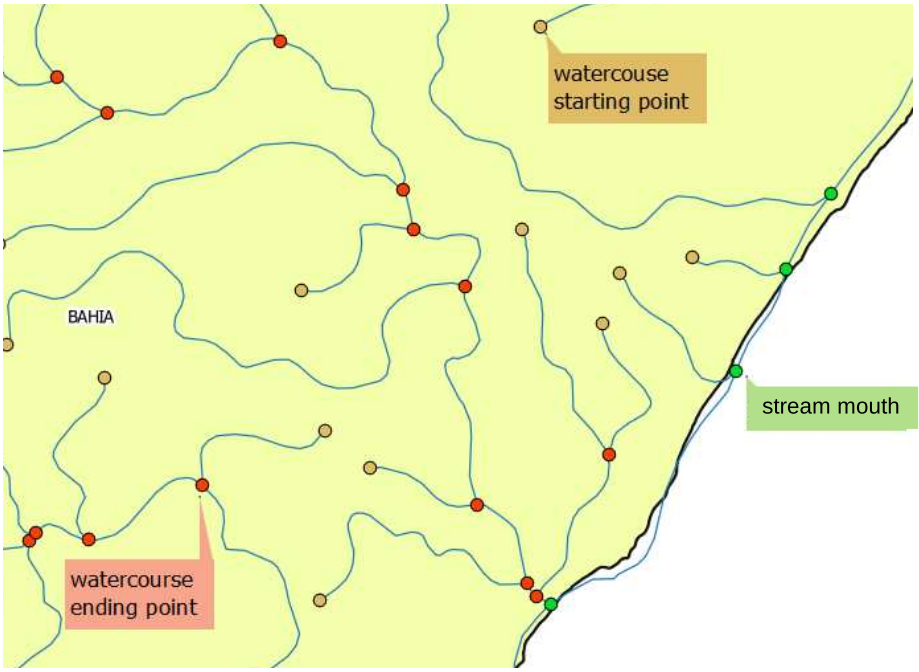


Figure 1. Kinds of Points in Drainage Network

The drainage stretches, on the other hand, represent only one geographic entity: the connection between two drainage points. Each stretch has two important attributes: (i) the hydronym, i.e., the name of the water body to which it belongs; and (ii) the hydrographic catchment area, which represents its importance in the drainage network – higher values indicate critical stretches with large areas of water catchment.

2.1.1. Cartographic Aspects

The scale of the Brazilian drainage network varies according to the cartographic mapping used as base in each geographic region, as shown in Figure 2. The Brazilian official

cartography, projected in the WGS84 Spatial Reference [1] is the start point of the mapping process. The steps of the hydrographic vectorization comprise the representation of each watercourse as a one-line entity, and identification of their crossing areas as start, end or confluence points. Digital elevation models (such as SRTM - Shuttle Radar Topography Mission [2]) are usually applied in the process of layout refine.

Research on specific watersheds is funded according to their strategic or economic importance, thus generating more detailed data in some regions. Figure 2 shows part of the drainage stretches in three scales: 1:1.000.000 (the majority of Brazilian watersheds), 1:250.000 (river Paraiba do Sul) and 1:50.000 (basin of rivers Piracicaba, Capivari and Jundiai) [3]. The latter, for instance, supplies one of Brazil's most populated regions and is the target of several studies, headed by the "PCJ Consortium". This consortium is composed by a group of cities and companies concerned about planning and financial support actions towards the recovery of water sources and raising societal awareness about the importance of watersource issues.

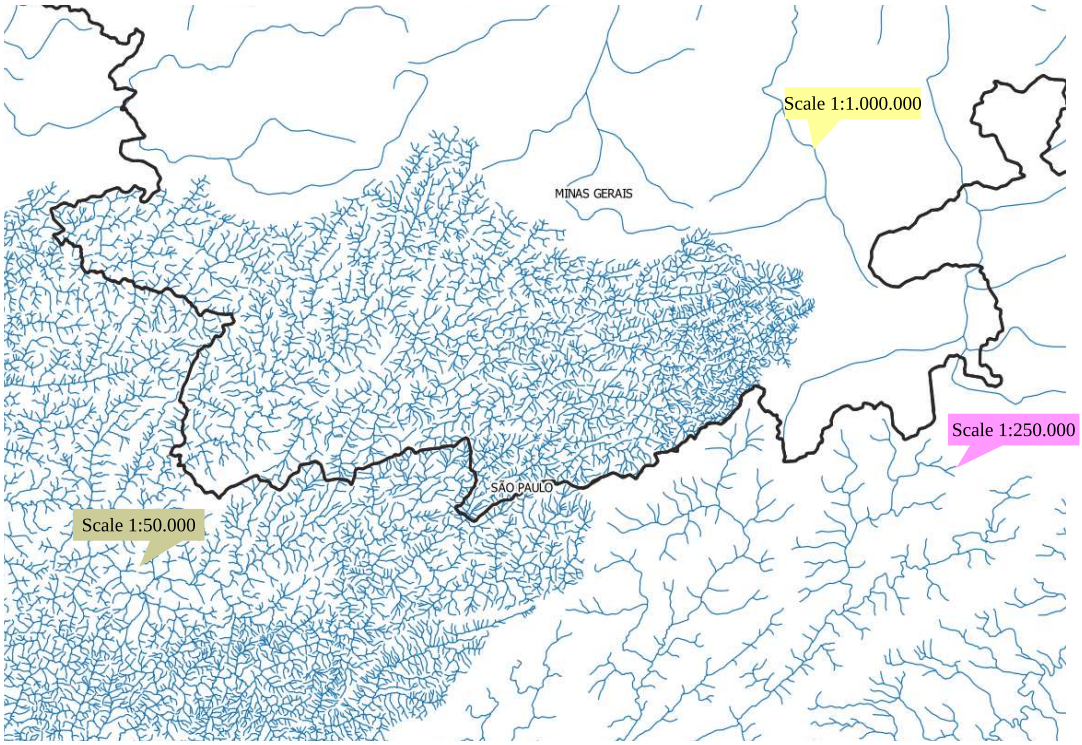


Figure 2. Different Drainage Stretch Scales in Drainage Network

The cartographic representation of the drainage network provides an important input to territorial analyses, i.e., when it is necessary to overlay the hydrographic data with other layers (using the geospatial information as the integrating component), in an attempt to understand some spatial phenomenon.

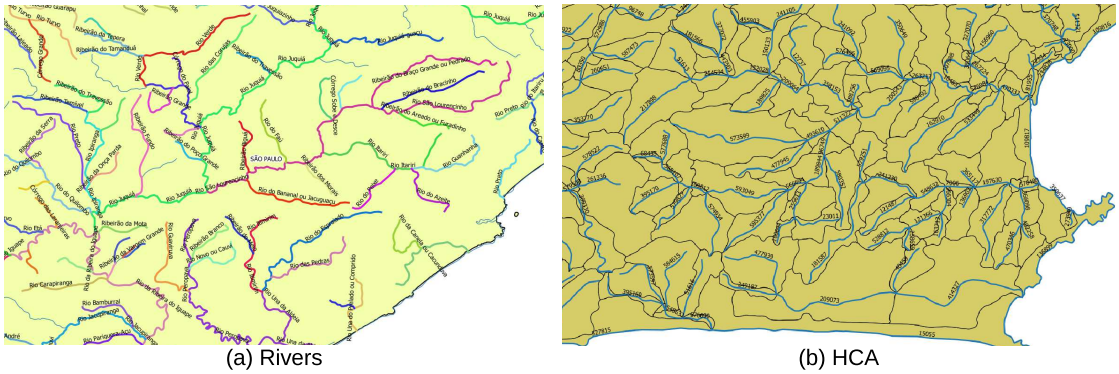
[1] spatialreference.org/ref/epsg/4326

[2] www2.jpl.nasa.gov/srtm

[3] Metadata available in: http://metadados.ana.gov.br/geonetwork/srv/pt/main.home?uuid=7bb15389-1016-4d5b-9480-5f1acdadd0f5

2.1.2. Logical Elements

There are at least four important logical elements in the Brazilian water resources database: hydronyms, hydrographic catchment areas, watersheds and main watercourses. The hydronym is an immutable attribute associated with each drainage stretch that indicates the logical element commonly known as “river”. A river is composed by all drainage stretches that are connected and have the same hydronym. Figure 3 (a) partially shows the drainage network under this perspective.



(a) Rivers (b) HCA

Figure 3. (a) Rivers: continuous drainage stretches with the same hydronym and (b) HCA: drainage stretches and their hydrographic catchment area

The other three elements are computed. Every time that the drainage network is updated these elements have to be recalculated. Updates occur for instance during some cartographic refinement process (more accurate scales) or to reflect human actions (e.g., by river transposition or construction of artificial channels). Updates do not occur very often. Thus, if the algorithms that construct the network are well defined, it is possible to materialize network elements, and update them whenever necessary.

The hydrographic catchment area (HCA) is a drainage stretch attribute, represented as a polygon, that delimits the water catchment area of the stretch. This delimitation is highly influenced by relief, given it influence in the water flow. Although HCA is a geospatial attribute, as shown in Figure 3 (b), only its area is relevant in most analyzes.

Watersheds and watercourses are two correlated elements – one is used to determine the other in a recursive way. A watershed is the logical element that delimits a drainage system channel. It is the official territorial unit for the management of water resources adopted by ANA. Unlike a basin – that refers only to where the water passes through – a watershed comprises the entire area that separates different water flowing. Every watershed has a main watercourse.

ANA adopts the Otto Pfafstetter Coding System [Pfafstetter 1989](ottocode) to define the watershed division process and watercourse identification. Each digit in the ottocode embeds a context about the stream (the main river or inter-basin, for instance). The main watercourse of a watershed is a set of connected drainage stretches selected by a traversal in the sub drainage network. It is constructed by selecting, in every confluence, the stretch with the largest hydrographic catchment accumulated area upstream (from the mouth to the spring). Following the watercourse layout, the watershed can be split in a set of sub-watersheds and the ottocode allows retrieving their hierarchical relations.

A $n - level$ watershed has a code with n digits. Figure 4 illustrates one step of this methodology: 4 (a) shows the drainage network of the watershed *Rio Trombetas* and its main watercourse, which has the ottocode 454 (level 3). Figure 4 (a) shows the 9 new watersheds created (level 4) by applying recursively the same methodology. The original code 454 is held as prefix to new watershed codes. More details about this methodology can be found in [Pfafstetter 1989].

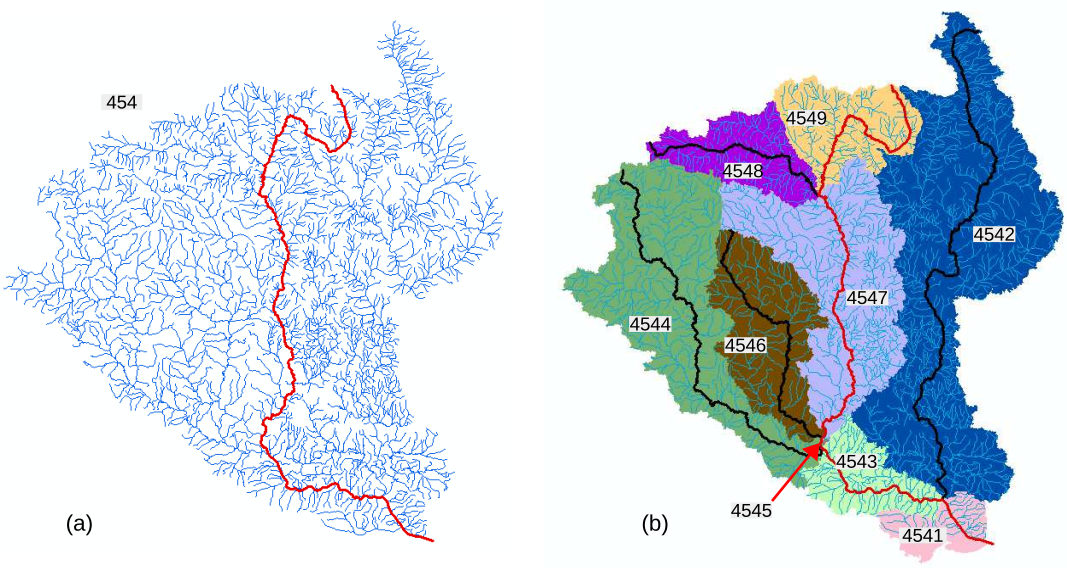


Figure 4. Otto Pfafstetter methodology

As can be seen, there are many studies that can take advantage of the network structure of this database and its logical preprocessed elements, even without considering geospatial aspects. Graph algorithms can be used, for instance, to ensure the network consistency or even to determine the main watercourse in a watershed; the latter can be found through a traversal algorithm in a subset of the drainage network, using higher HCA values as the navigation criterion.

2.2. Graph Data Management Paradigm

The graph data management paradigm is characterized by using graphs (or their generalizations) as data models and graph-based operations to express data manipulation. It is relationship driven, as opposed to the relational data model which requires the use of foreign keys and joins to infer connections between data items. Graph databases are usually adopted to represent data sets where relations among data and the data itself are at the same importance level [Angles and Gutierrez 2008]. Graph data models appeared in the 90's; nevertheless, only in the past few years have they been applied to information management systems, propelled by the rise of social networks such as Facebook and Twitter.

The formal foundation of all graph data models is based on variations on the mathematical definition of a graph. In its simplest form, a graph G is a data structure composed by a pair (V, E), where V is a finite non empty set of vertices and E is a finite set of edges connecting pairs of vertices. On top of this basic layer, several graph data structures were proposed by the database community, attempt to improve expressiveness, representing data in a better (and less ambiguous) way, such as property

graph (or attributed graph) [Rodriguez and Neubauer 2010, Robinson et al. 2013], hypernode [Levene and Loizou 1995] and RDF graph [Bonstrom et al. 2003].

Considering the edges, a graph can be directed (i.e., there is a tail and head to each edge); single relational or multi-relational (i.e., multiple relationships can exist between two vertices). The connection structure affects the traversal. An edge can have different meanings, such as attributes, hierarchies or neighborhood relations. Despite their flexibility and efficient management of heavily linked data, there is no consensual data structure and query language for graph databases.

One of the most popular graph structures is the property graph (or attributed graph) [Rodriguez and Neubauer 2010, Robinson et al. 2013]. It tries to arrange vertex and edge features in a flexible structure through key-value pairs (e.g., type, label or direction).

3. Implementation

3.1. Original Relational Database: pgHydro

The pgHydro project [4] – developed by ANA and started in 2012 – aims to implement a spatial relational database to manage the hydrographic objects that compose the Brazilian Water Resources database [Teixeira et al. 2013]. It encompasses tables, constraints and views, and a set of stored procedures to ensure data consistency and to process routine calculations. The conceptual model of pgHydro is illustrated in Figure 5.

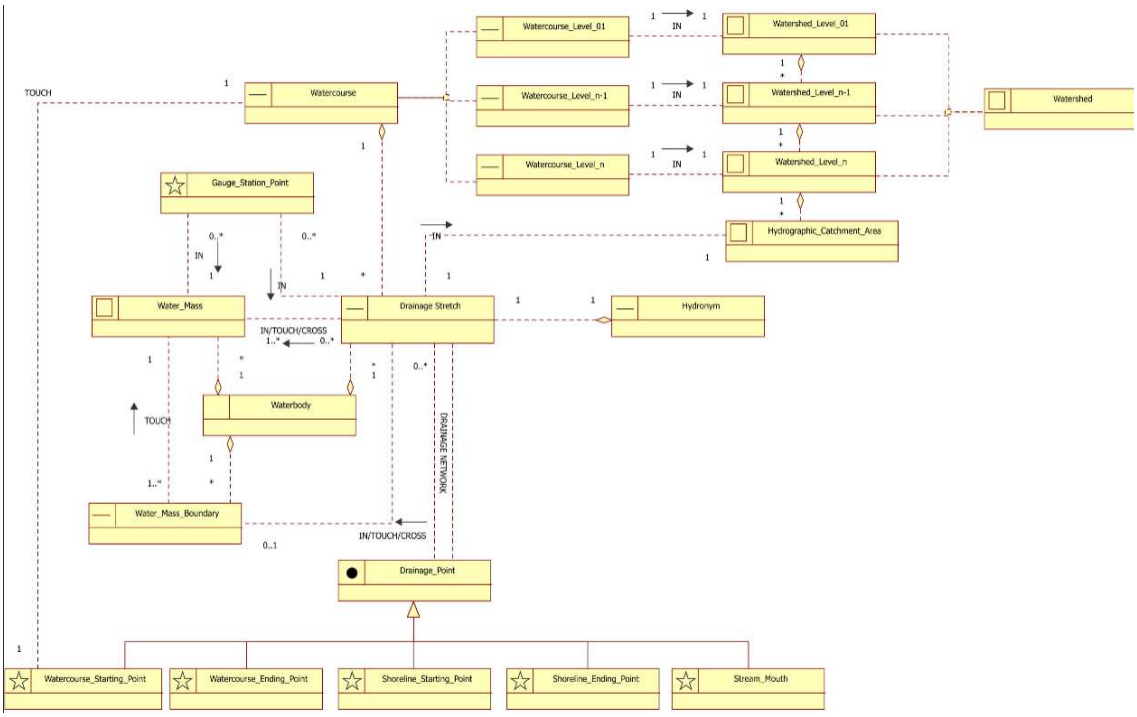


Figure 5. PgHydro Database Conceptual Model

PgHydro was implemented in PostGIS/PostgreSQL and a Python interface. PgHydro is a free and open source project and is available for companies and organizations with

[4] pghydro.org

an interest in management and decision making in water resources. More spatial analysis can be done using GIS, such as ArcGIS [5] or QuantumGIS [6].

3.2. Proposal Graph Database: HydroGraph

We have transformed ANA relational database (the drainage network) into a graph database, here denoted by G_{Hydro} (partially illustrated in Figure 6), keeping the same basic structure of vertices (the drainage points) and edges (the drainage stretches). This data model makes easier to understand the drainage network as it really is: a binary tree-graph, connected and acyclic, whose edges go from the leaves to the root.

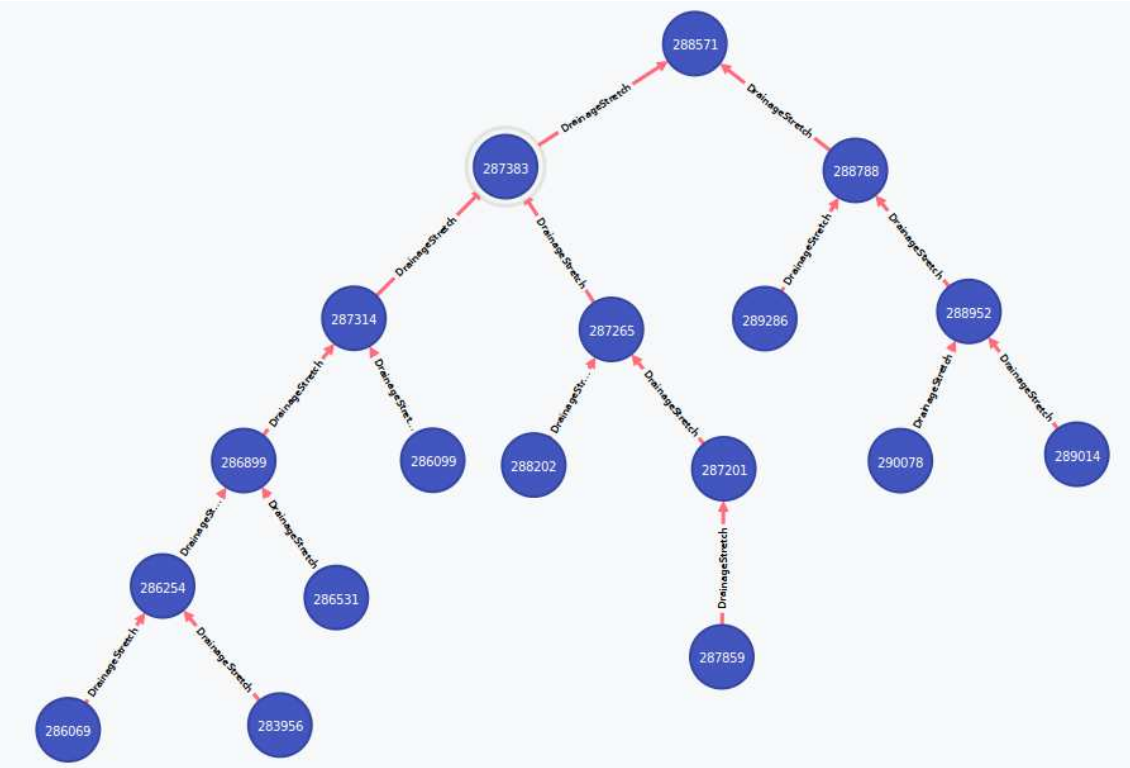


Figure 6. G_{Hydro}: Brazilian Drainage Network as a Graph Database

The graph database chosen was Neo4j [7] – a labeled property multi-graph [Robinson et al. 2013]. Every edge must have a relationship type, and there is no restrictions about the number of edges between two nodes. Both vertices and edges can have properties (key-value pairs) and index mechanism. Neo4j implements a native disk-based storage manager for graphs, a framework for graph traversals and an object-oriented API for Java. It is an open source project and it is nowadays the most popular graph database [8].

The creation and population of G_{Hydro} were done through **LOAD CSV** command – a load engine provided by Neo4j. The input could be a local or a remote classical CSV

[5] www.arcgis.com
[6] www.qgis.org
[7] neo4j.com
[8] According to DB-Engines Ranking of Graph DBMS (accessed on September, 2015) [db-engines.com/en/ranking/graph+dbms]

file – containing a header and a set of lines in which each line represents a record, and the line is a set of fields separated by comma. The CSV files were extracted from the PostgreSQL database using the **COPY** command [9]. Figure 7 shows some of the LOAD CSV commands that giving rise to G_{Hydro} (commands (i) to drainage points and (iii) to drainage stretches). Commands (ii) and (iv) ensure the integrity constraint of unique values for all the identifiers.

(i)
```
LOAD CSV WITH HEADERS FROM "file:<path>/drainage_point.csv" AS line
CREATE (p:DrainagePoint {id:toInt(line.id), valence:toInt(line.valence), geom:line.geom})
```

(ii)
```
$ CREATE CONSTRAINT ON (point:DrainagePoint) ASSERT point.id IS UNIQUE
```

(iii)
```
$ LOAD CSV WITH HEADERS FROM "file:<path>/drainage_stretch.csv" AS line match
(source:DrainagePoint { id: toInt(line.drs_drp_pk_sourcenode)}),
(target:DrainagePoint { id: toInt(line.drs_drp_pk_targetnode)}) CREATE (source)-
[:DrainageStretch { id: toInt(line.drs_pk), upstreamstretch: toInt
(line.drs_drs_pk_upstreamstretch), downstreamstretch: toInt
(line.drs_drs_pk_downstreamstretch), distancetosea: line.drs_nu_distancetosea,
distancetowatercourse: line.drs_nu_distancetowatercourse, waterbodyoriginal:
line.drs_nm_waterbodyoriginal, length: line.drs_gm_length, hca:
line.drs_hca_pk,  upstreamarea: line.drs_nu_upstreamarea, wtc: line.drs_wtc_pk,
hdr: line.drs_hdr_pk, geom: line.st_astext, domain:line.drs_wtc_ds_domain
}]->(target)
```

(iv)
```
$ CREATE CONSTRAINT ON (stretch:DrainageStretch) ASSERT stretch.id IS UNIQUE
```

Figure 7. LOAD CSV commands

The **LOAD CSV** command is based on Cypher syntax, the graph query language available on Neo4j [Robinson et al. 2013]. Cypher is a pattern oriented, declarative query language. It has two kinds of query structures: a read and a write query structure. The pattern representation is inspired by traditional graph representation of circles and arrows. Vertex patterns are represented in parenthesis; and edge patterns in brackets between hyphens, one of which with a right angle bracket to indicate the edge direction. For example, the expression **(a)-[r:RELATED]->(b)** is interpreted as two vertex patterns **a** and **b** and one edge pattern **r**, type **RELATED**, that starts on vertex **a** and ends in vertex **b**.

3.3. PgHydro Functions

The most important functions of pgHydro are:

1. To validate drainage network consistent;
2. To define the direction of water flow;
3. To apply Otto Pfafstetter's watershed coding system;
4. To select the set of upstream/downstream stretches;
5. To calculate the upstream hydrographic/downstream catchment area.

As can be readily seen, most of these functions can be solved applying to graph algorithms on G_{Hydro}. Execute these tasks over relational databases would require many join operations – one of the most computationally expensive processes in SQL databases. Another possibility would be to build an in-memory network representation on top of the

[9] www.postgresql.org/docs/9.2/static/sql-copy.html

relational storage model and to use APIs and programming languages. Graph databases exempt the need of intermediate models from storage to application logic layer.

Consistency tests over the drainage network concern mainly two aspects: connectivity of all stretches and the binary tree structure. In graph terms – considering G_{Hydro} implementation – we can apply the connected component analysis solution. A connected component in a graph G is a subgraph H of G in which, for each pair of vertices u and v, there is a path connecting u and v. If more than one connected component is found in G_{Hydro}, the database is inconsistent. The binary tree structure, on the other hand, is checked selecting all vertices whose degree value are different from 1 (start or end points) or 3 (confluences).

The selection of the upstream stretches can be done applying to Depth-First Search, starting on the stretch of interest and ending on the watershed root. To calculate the upstream hydrographic catchment area, we sum the HCA from each drainage stretch returned in the previous selection. The same approach can be applied to downstream stretches, using the opposite navigation direction and aggregating all subtrees.

The calculation of the Otto Pfafstetter watershed coding is a more complex task, but it is still a graph traversal. The base task is to define the main watercourse. Here, unlike the previous computations we need to establish graph traversal criteria on each node: selecting, at every confluence, the stretch with the largest HCA accumulated upstream.

Among all these functions, only the definition of water flow direction is actually a GIS task and depends on the geospacial information. This calculation involves solving equations that examine the relationship among several variables such as stream length, water depth, resistance of the surface and relief.

4. Research Challenges

There are at least three important challenges involved in our approach. The first is related to the incompleteness of graph data models. According to the classical definition, a complete data model should be composed by three main elements: (i) data structure types, (ii) operators to retrieve or derive data and (iii) integrity rules to define consistent the database states [Codd 1980]. Related work on graph data models shows that they are incomplete concerning least one of these aspects. Most of them concern only data structures – hypergraphs, RDF or property graphs. Others describe only query languages or APIs to manipulate or retrieve data. There are few attempts to discuss consistency or ACID properties over graph data models. This scenario hampers the formalizing of a complete graph data model. Besides, most implementations of graph databases do not adhere to the theoretical models.

Second, traditional Relational Database Management Systems (RDBMS) are the most mature solution to data persistence and usually the best option when strong consistency is required. Besides, there are many spatial extensions over RDBMS current used as foundation to geospatial systems and services. Therefore, in some cases there is need for the coexistence of both models – relational and graph – dividing tasks of management and analysis according to their specialties. This requires the development of hybrid architecture to enable the integration of relational and graph databases, as proposed by [Cavoto and Santanche 2015].

Finally, the task of network-driven analysis is not completely solved once the graph database is available. The graph data design (i.e., which data is represented as vertices, which is represented as edges and what kind of properties they have) can streamline or even render non-viably the extraction of topological or graph properties. There is no simple way to crossing through different designs in graph databases. This challenge is also goal of our research, as described in [Daltio and Medeiros 2014]. The idea is to specify and implement an extension of the concept of view (from relational databases) to graph database, thereby allowing managing and analyze a graph database under arbitrary perspectives. Consider this specific database, it would be possible to explore not only the drainage network, but also the network among the logical elements – rivers, watersheds and watercourses.

5. Conclusions

This paper presented our ongoing work to construct a graph database infrastructure to support analysis operations on the Brazilian Water Resources database. Our research shows the importance of graph driven analysis over the drainage network, rather than the computationally expensive process of relational databases for such analysis. It was presented the G_{Hydro} – a version of the original relational database implemented on Neo4j, composed by 620.280 drainage points (vertices) and 620.279 drainage stretches (edges).

Our research takes advantage of graph structures to model and navigate through relationships across the network and its logical elements – watersheds and watercourses. This helps analysts' work in analysis and forecast. However, given the complexity of geospatial data – mainly on big data proportions – there is still no single solution to solve all persistence, management and analysis issues. Hybrid architecture approaches seem to be the most flexible and complete choice.

Acknowledgment

Work partially financed by FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the Microsoft Research FAPESP Virtual Institute (NavScales project), FAPESP-PRONEX (eScience project), INCT in Web Science, and individual grants from CNPq. The authors thank Alexandre de Amorim Teixeira, geoprocessing expert of National Water Agency (ANA), for his help in clarifying watershed data analysis and relationship issues.

References

Amirian, P., Basiri, A., and Winstanley, A. (2013). Efficient online sharing of geospatial big data using nosql xml databases. In *Proceedings of the 2013 Fourth International Conference on Computing for Geospatial Research and Application*, COMGEO '13, pages 152–, Washington, DC, USA. IEEE Computer Society.

Amirian, P., Basiri, A., and Winstanley, A. (2014). Evaluation of data management systems for geospatial big data. In *Computational Science and Its Applications - ICCSA 2014*, volume 8583 of *Lecture Notes in Computer Science*, pages 678–690. Springer International Publishing.

Angles, R. and Gutierrez, C. (2008). Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39.

Bonstrom, V., Hinze, A., and Schweppe, H. (2003). Storing rdf as a graph. In *Web Congress, 2003. Proceedings. First Latin American*, pages 27–36.

Brazil (1997). Water resources federal (9.433). Federal Register [of] Federative Republic of Brazil, Executive Branch. Section 1, p. 470.

Brebbia, C. and Popov, V. (2011). *Water Resources Management VI*. WIT transactions on ecology and the environment. WIT Press.

Cavoto, P. and Santanche, A. (2015). Fishgraph: A network-driven data analysis. In *Proceedings of the 11th IEEE International Conference on eScience*, pages 1–10, Munich, Germany.

Codd, E. F. (1980). Data models in database management. *SIGMOD Rec.*, 11(2):112–114.

Daltio, J. and Medeiros, C. B. (2014). Handling multiple foci in graph databases. In Switzerland, S. I. P., editor, *Lecture Notes in Bioinformatics (LNBI) - Proceedings of 10th International Conference on Data Integration in the Life Sciences*, volume 8574, pages 58–65, Lisboa, Portugal.

Fry, B. J. (2004). *Computational Information Design*. PhD thesis, MIT. AAI0806331.

Hecht, R. and Jablonski, S. (2011). Nosql evaluation: A use case oriented survey. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 336–341.

Hey, T., Tansley, S., and Tolle, K., editors (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, Redmond, Washington.

Levene, M. and Loizou, G. (1995). A graph-based data model and its ramifications. *IEEE Trans. Knowl. Data Eng.*, 7(5):809–823.

Pfafstetter, O. (1989). Watershed classification: coding methodology (in portuguese). National Department of Sanitation Construction. Rio de Janeiro, RJ.

Robinson, I., Webber, J., and Eifrem, E. (2013). *Graph Databases*. O'Reilly Media, Incorporated.

Rodriguez, M. A. and Neubauer, P. (2010). Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41.

Teixeira, A. A., Silva, A. M., Molleri, G. S. F., Ferreira, F. V., and Borelli, A. J. (2013). Pghydro - hydrographic objects in geographical database (in portugueses). In *Proceedings of the 2013 Brazilian Symposium on Water Resources*, pages 1–8.