Brazilian Journal of Cartography (2016), Nº 68/6, Special Issue GEOINFO 2015: 1181-1189
Brazilian Society of Cartography, Geodesy, Photgrammetry and Remote Sense
ISSN: 1808-0936

HYDROGRAPH: EXPLORING GEOGRAPHIC DATA IN GRAPH DATABASES

HYDROGRAPH: Explorando Dados Geográficos em Bancos de Dados de Grafos

Jaudete Daltio & Claudia M. Bauzer Medeiros

University of Campinas - UNICAMP
Institute of Computing
Avenida Albert Einstein, 1251, Cidade Universitária, Campinas/SP - Brazil
{jaudete, cmbm}@ic.unicamp.br

Received on February 28, 2015/ Accepted on March 29, 2016
Recebido em 28 de Fevereiro, 2015/ Aceito em 29 de Março, 2016

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.

Keywords: Graph Databases, Watershed, Geographic Data.

RESUMO

A cada dia, a disponibilidade de água potável se torna mais escassa. Informações confiáveis sobre o volume de água e sua qualidade em cada bacia hidrográfica são cada vez mais importantes no processo de gestão e planeamento do uso sustentável e consciente do uso da água. Metodologias de pesquisa voltadas para o uso intensivo de dados estão sendo cada vez mais utilizadas nesse tipo de contexto. Análises neste cenário demandam, por exemplo, a manipulação de grandes redes de drenagem e suas bacias hidrográficas. O trabalho em andamento apresentado neste artigo explora dados de bacias geográficas utilizando bancos de dados de grafos – um tipo de banco de dados *NoSQL* escalável e flexível. O banco de dados das Bacias Hidrográficas brasileiras foi utilizado como estudo de caso. O mapeamento entre o modelo geográfico e o modelo de grafos é baseado na estrutura de conexões que emerge de forma natural a partir das relações topológicas entre entidades geográficas.

Palavras-chave: Banco de Dados de Grafos, Bacias Hidrográficas, Dados Geográficos.

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 & 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 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 several aspects.

Geospatial data about water resources fits these graph connectivity criteria. 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 data storage 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 the specification of important queries over watersheds in graph terms. Some research challenges involved are presented in section 4. Finally, section 5 presents conclusions and ongoing work.

2. RESEARCH SCENARIO AND THEORETICAL FOUNDATIONS

The research scenario that underpins this work is the management of geospatial watershed data as graphs. Thereby, two important topics are described in this section: the *Brazilian* watershed data with it´s different perspectives and the concept of graphs as a paradigm to manage data.

2.1 Brazilian Water Resources Database

In terms of water resources, Brazil is a privileged country: it holds 12% of the world total and the largest reserve of fresh water on Earth (BREBBIA & 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, 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) 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 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:

(a) a watercourse start point, usually a spring or water source;

(b) a watercourse end point, usually a river mouth;

(c) a stream mouth point, which flows into the sea;

(d) 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.

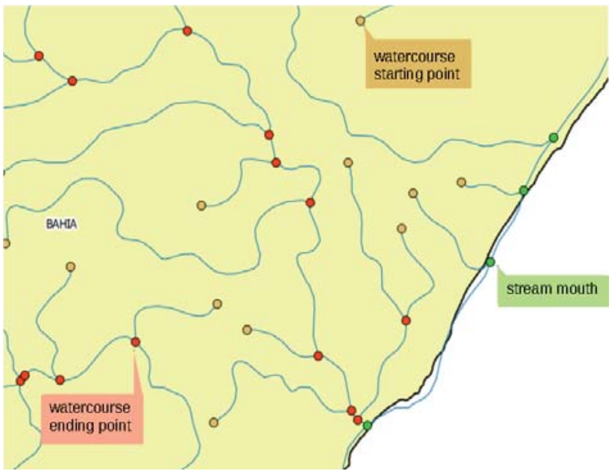

Fig. 1 - Kinds of *Points* in Drainage Network.

The drainage *stretches*, on the other hand, represent only one geographic entity: the connection between two drainage *points*. Each stretch has two important attributes: the hydronym, i.e., the name of the water body to which it belongs; and the hydrographic catchment area (HCA), 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. HCA value determines the importance of a drainage stretch in the drainage network – higher values indicate critical *stretches* with large areas of water catchment. Although HCA is a geospatial attribute, as shown in Figure 2, only its area is relevant in most analyzes.

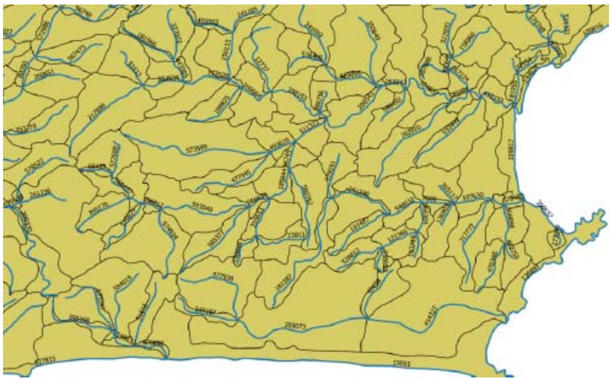

Fig. 2 - HCA: drainage *stretches* and their hydrographic catchment area.

Based on these attributes, there are at least tree important logical elements over the drainage network: rivers, 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 partially shows the drainage network under this perspective.

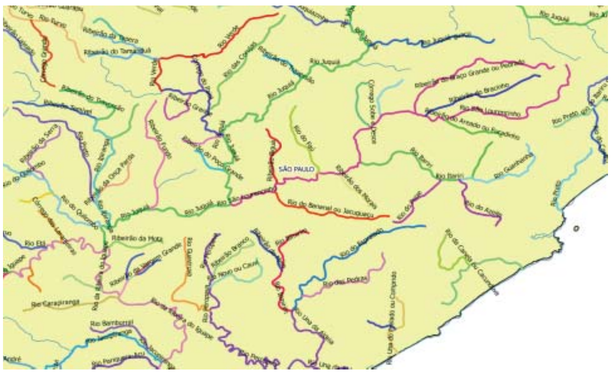

Fig. 3 - Rivers: continuous drainage *stretches* with the same hydronym.

Watersheds and watercourses are two correlated elements – one is used to determine the other in a recursive way. A watershed is composed by a set of hydrographic catchment area (dissolved) and 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.

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). Every watershed has a main watercourse - a set of connected drainage *stretches* selected by a traversal in the sub drainage network.

Every time that the drainage network is updated, watersheds and watercourses 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).

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 4. The *Brazilian* official cartography, projected in the WGS84 Spatial Reference is the start point of the mapping process *(spatialreference.org/ref/epsg/4326)*. The steps of the hydrographic vectorization comprise the representation of each watercourse as a single line (in terms of geographic entity) and the identification of their crossing areas as start, end or confluence *points*. Digital elevation models (such as SRTM - Shuttle Radar Topography Mission - *www2.jpl.nasa.gov/srtm*) are usually applied in the refine process.

Research on specific watersheds is funded according to their strategic or economic importance, thus generating more detailed data in some regions. Figure 3 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) (*Metadata available in: http://metadados.ana.gov.br/geonetwork/ srv/pt/main.home?uuid=7bb15389-1016-4d5b-9480-5f1acdadd0f5*). 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 society awareness about the importance of water source issues.

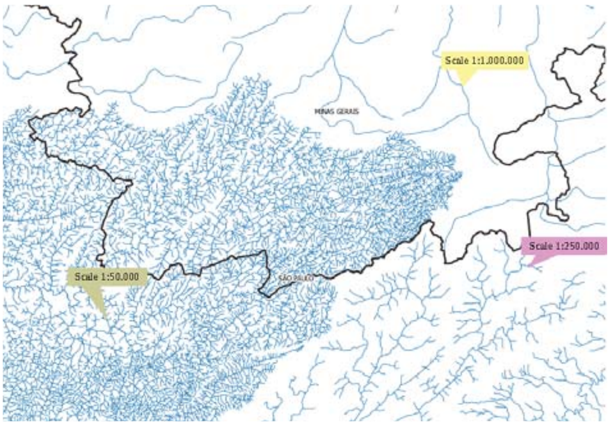

Fig. 4 - Different Drainage Stretch Scales in Drainage Network;

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.

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. In this model, queries are performed through graph traversals, pattern matching or graph algorithms. Graph databases are usually adopted to represent data sets where relations among data and the data itself are at the same importance level (ANGLES & GUTIERREZ, 2008).

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.

Considering the edges, a graph can be directed (i.e., there is a source and target 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. The more popular graph structures are hypernode (LEVENE & LOIZOU, 1995), RDF graph (BONSTROM *et al*., 2003) and property (RODRIGUEZ & NEUBAUER, 2010). The last, adopted in our research, tries to arrange vertex and edge features in a flexible structure through key-value pairs (e.g., type, label or direction) (ROBINSON *et al*., 2013).

3. IMPLEMENTATION

The solution implementation started with the conversion of the original watershed relational database into a graph database. Afterwards, the main functions commonly used to data exploration were mapped to graph algorithms according to the new database structure. These steps are detailed presented in next sections.

3.1 Original Relational Database: pgHydro

The pgHydro project (*pghydro.org*) – developed by ANA and started in 2012 – implements 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. As can be seen, Drainage Stretch, Drainage Point, Watershed and Watercourse are the main entities and a Drainage Stretch is defined as a pair of Drainage *Points*.

PgHydro was implemented in PostGIS/PostgreSQL as 26 tables, 25 views and 137 stored procedures. PgHydro is a free and open source project and is available for companies and organizations with an interest in management and decision making in water resources. *PgHydro* has also a *Python* interface with the most common queries. More spatial analysis can be done using GIS, such as *ArcGIS* or *QuantumGIS*.

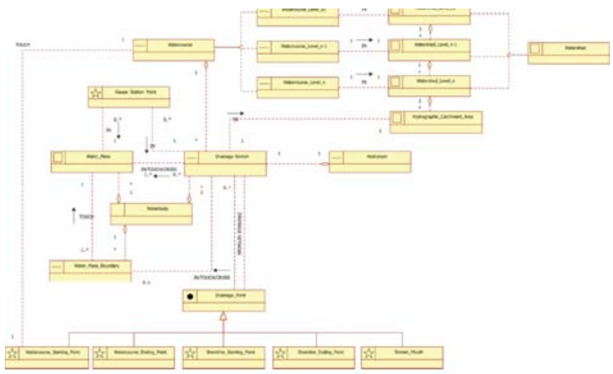

Fig. 5 - PgHydro Database Conceptual Model.

3.2 Proposal Graph Database: HydroGraph

We migrated ANA relational database into a graph database, here denoted by GHydro (partially illustrated in Figure 6). GHydro keeps the same basic structure of vertices (the drainage *points*) and edges (the drainage *stretches*) of the original drainage network. The first advantage was to get a real understanding about what the drainage network really is: a binary tree graph, connected and acyclic, whose edges go from the leaves to the root.

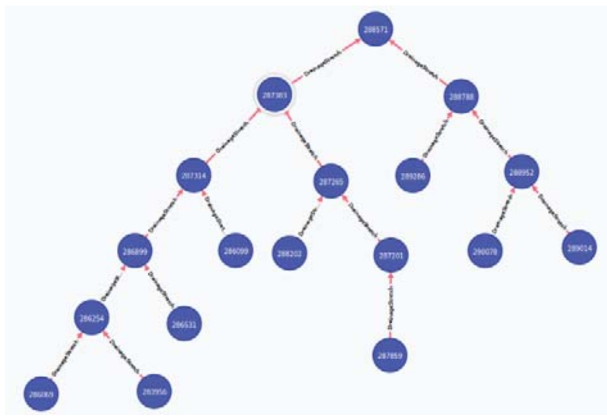

Fig. 6 - GHydro: Brazilian Drainage Network as a Graph Database.

The graph database chosen was *Neo4j* – a labeled property multigraph (ROBINSON *et al.*, 2013). *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 (*According to DB-Engines Ranking of Graph DBMS, accessed on September, 2015*).

Neo4j demands that every edge must have a relationship type – in our model all edges have type *Drainage_Stretch.* . All vertices are labeled as *Drainage_Point*. All the original attributes of both entities were migrated as properties.

Neo4j provides a language, named *Cypher*, to query and manipulate data (ROBINSON *et al.*, 2013). *Cypher* is a pattern oriented, declarative query language. 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.

The creation and population of G_{Hydro} were done through LOAD CSV command – a load engine provided by *Neo4j* based on *Cypher* syntax. The input is a classical CSV file: 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 original relational *PostgreSQL* database of *PgHydro* using the COPY command.

Figure 7 shows part of the LOAD CSV commands that give rise to GHydro (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.

3.3 PgHydro Functions

Once G_{Hydro} is created, we can redefine the more important functions of *pgHydro*, starting from the consistency tests. Consistency tests over the drainage network concern mainly two aspects: (i) connectivity of all *stretches* and (ii) the binary tree structure. To check the first aspect, 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. The binary tree structure (item (ii)), on the other hand, is checked selecting all vertices whose degree value are different from 1 (start or end *points*) or 3 (confluences). The database is inconsistent if G_{Hydro} has at least one vertex with this property or if more than one connected component were found.

```
(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
```

Fig. 7 - LOAD CSV commands.

An important function is to retrieve the upstream *stretches* of a drainage point. In GHydro this can be done applying a Depth-First Search algorithm, starting on the drainage point of interest and ending on the graph root (or the root of a watershed). Similarly, it is possible to calculate the upstream hydrographic catchment area (HCA): the sum of the HCA attribute from each *Drainage Stretch* returned in the previous selection.

Although rivers are the most popular concept, the official territorial unit adopted by ANA for management tasks is the watershed. The calculation of a watershed ottocode is also based on traversal in the sub drainage network, where the goal is to define the main watercourse. The main watercourse is constructed by selecting, in every confluence, the stretch with the largest accumulated HCA 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 8 illustrates one step of this methodology: 8 (a) shows the drainage network of the watershed *Rio Trombetas* and its main watercourse, which has the ottocode 454 (level 3). Figure 8 (b) 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).

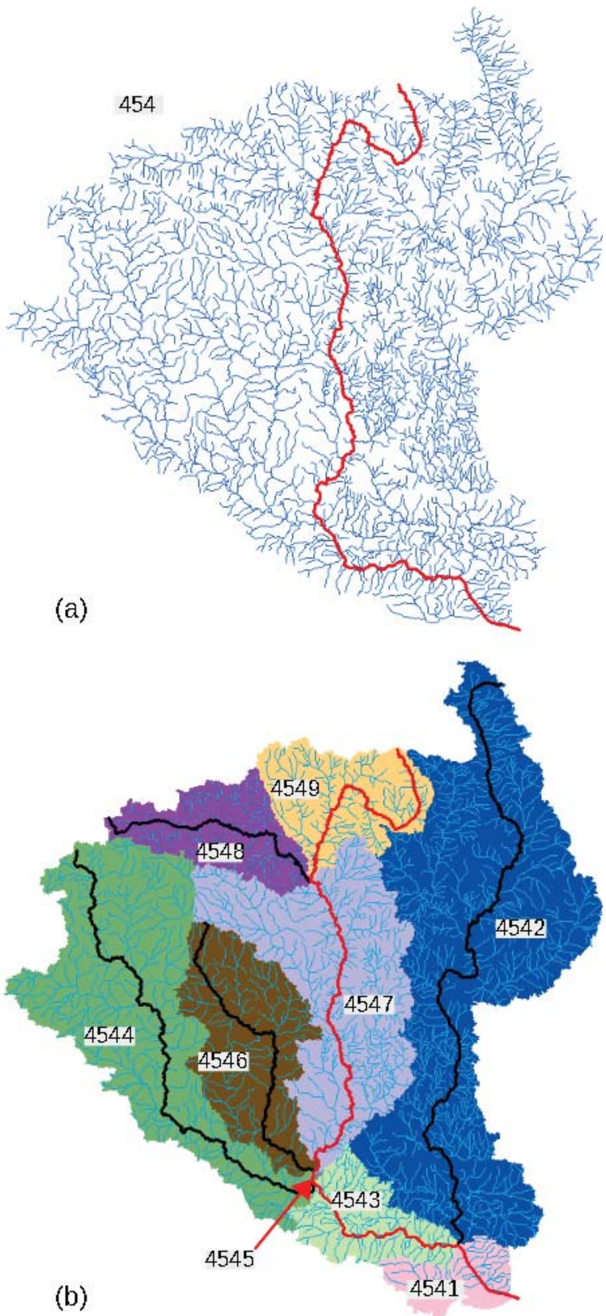

Fig. 8 - Otto Pfafstetter methodology

As can be readily seen, these functions can be solved applying graph algorithms to GHydro. The execution of these tasks over the original 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 relational storage model and to use APIs and programming languages. Graph databases avoid the need for intermediate models from storage to application logic layer.

Another observation here is that 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. Among the more important functions, the definition of water flow direction is actually a GIS task and totally dependent on the geospatial information. This calculation involves solving equations that examine the relationships 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 (CODD, 1981), 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. Related work on graph data models is 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 formalization 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 both models may have to be used concomitantly – relational and graph – splitting tasks of management and analysis according to their specialties. This requires the development of a hybrid architecture to enable the integration of relational and graph databases, as proposed by (CAVOTO & SANTANCHE, 2015).

Finally, the task of network-driven analysis is not completely solved once the data is stored in a graph database. 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-viable the extraction of topological or graph properties. There is no simple way to cross through different designs in graph databases. This challenge is also a goal of our research, as described in (DALTIO & MEDEIROS, 2016). The idea is to specify and implement an extension of the concept of view (from relational databases) to graph databases, 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. The paper presented G_{Hydro} that restructures 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. This helps analysts' work in analysis and forecast. However, given the inherent complexity of geospatial data, there are multiple possible graph designs to represent it. This scenario leads to new research challenges to allow extract different logical elements on graph databases – in our case, rivers, watersheds and watercourses over drainage network.

ACKNOWLEDGMENT

Work partially financed by FAPESP/ Cepid in Computational Engineering and Sciences (2013/08293-7), 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. & WINSTANLEY, A. Efficient Online Sharing of Geospatial Big Data Using NoSQL XML Databases. In: **Fourth International Conference on Computing for Geospatial Research and Application (COMGEO' 13).** Proceedings, IEEE Computer Society. Washington, DC, USA. pp 152. 2013.

AMIRIAN, P.; BASIRI, A. & WINSTANLEY, A. Evaluation of Data Management Systems for Geospatial Big Data. In: **Computational Science and Its Applications (ICCSA 2014).** Proceedings, Lecture Notes in Computer Science, volume 8583, Springer International Publishing. pp 678–690. 2014.

ANGLES, R. & GUTIERREZ, C. Survey of Graph Database Models. In: **ACM Computing Surveys**, volume 40, issue 1. pp 1-39. 2008.

BONSTROM, V.; HINZE, A. & SCHWEPPE, H. Storing RDF as a graph. In: **First Latin American Web Congress (2003).** Proceedings. pp 27–36. 2003.

BRAZIL. Brazilian Water Law, no. 9.433. **Federal Register [of] Federative Republic of Brazil, Executive Branch**. Section 1. pp 470. 1997.

BREBBIA, C. & POPOV, V. Water Resources Management VI. **WIT Transactions on Ecology and the Environment**. WIT Press. 864 p. 2011.

CAVOTO, P. & SANTANCHE, A. **FishGraph: A Network-driven Data Analysis.** In: **11th IEEE International Conference on eScience.** Proceedings, Munich, Germany. pp 1–10. 2015.

CODD, E. F. Data Models in Database Management. ACM **SIGMOD Record, volume 11, issue 2.** pp 112–114. 1981.

DALTIO, J. & MEDEIROS, C. B. HydroGraph: Exploring Geographic Data in Graph Databases. In: **XVI Brazilian Symposium on Geoinformatics (GEOINFO 2015)**. Proceedings. pp 44–55. 2015.

DALTIO, J. & MEDEIROS, C. B. A View Handler for Semantic Graphs. In: **10th IEEE International Conference on Semantic Computing (ICSC 2016).** Proceedings, IEEE Conference Publications. Los Angeles, USA. pp 1–5. 2016.

FRY, B. J. Computational Information Design. **PhD Thesis, Massachusetts Institute of Technology (MIT), School of Architecture and Planning**. 175 p. 2004.

HECHT, R. & JABLONSKI, S. NoSQL evaluation: A use case oriented survey. In: **International Conference on Cloud and Service Computing (CSC 2011)**. Proceedings, Hong Kong. pp 336–341. 2011.

HEY, T.; TANSLEY, S. & TOLLE, K. The Fourth Paradigm: Data-Intensive Scientific Discovery. **Microsoft Research, 1º edition.** Redmond, Washington. 284 p. 2009.

LEVENE, M. & LOIZOU, G. A graph-based data model and its ramifications**.** In: **Journal IEEE Transactions on Knowledge and Data Engineering.** Volume 7, issue 5. pp 809–823. 1995.

PFAFSTETTER, O. Watershed Classification: Coding Methodology (in portuguese). In: **National Department of Sanitation Construction. Draft.** Rio de Janeiro. 19 p. 1989.

ROBINSON, I.; WEBBER, J. & EIFREM, E.