A First Approach towards Storage and Query Processing of Big Spatial Networks in Scalable and Distributed Systems

Manel Mena*, Antonio Corral*, and Luis Iribarne*

Dept. of Informatics, University of Almeria, Almeria, Spain.
E-mail: {manel.mena,acorral,luis.iribarne}@ual.es

Abstract. Due to the ubiquitous use of spatial data applications and the large amounts of spatial data that these applications generate, the processing of large-scale queries in distributed systems is becoming increasingly popular. Complex spatial systems are very often organized under the form of *Spatial Networks*, a type of graph where nodes and edges are embedded in space. Examples of these spatial networks are transportation and mobility networks, mobile phone networks, social and contact networks, etc. When these spatial networks are big enough that exceed the capacity of commonly-used spatial computing technologies, we have *Big Spatial Networks*, and to manage them is necessary the use of distributed graph-parallel systems. In this paper, we describe our emerging work concerning the design of new storage methods and query processing algorithms over big spatial networks in scalable and distributed systems, which is a very active research area in the past years.

Keywords: Spatial Networks, Storage Methods, Query Processing, Distributed Systems

1 Introduction

Spatial Computing represents ideas, solutions, tools, technologies and systems that continuously transform our lives and society through the creation of a new understanding of spaces, their locations and properties [5]. This spatial information allows us to know aspects of how do we communicate, how do we visualize our relation to places in a space of interest, how can we navigate through those places, etc. For this reason, this area is specially interesting at research level.

Big Data comes with a series of inherited challenges like how do we collect big data, the need of real time processing, ways to find hidden information, recurrent patterns and new connections between data, etc. [10]. *Spatial data* consists of points, lines, polygons and other geographic and geometric data primitives, which can be mapped by location in a geographic coordinate system. There are three basic models to represent spatial data: raster (e.g. satellite images), vector

* Work funded by the MINECO research project [TIN2017-83964-R] and Manel Mena by a Grant of PPIT-2017 Ual

(e.g. points, lines, polygons, etc.), and network (e.g. spatial networks). In [6], the concept of *Spatial Big Data* (SBD) is defined as instances of these spatial data types that exhibit at least two of the 3 V's: Volume, Velocity, and Variety. SBD represents the next frontier in spatial computing. Some examples of emerging SBD include information about traffic speed in transportation networks, GPS trajectory data, gas emission in a road networks, etc.

Big Spatial Networks (BSN) refers to *Spatial Networks* (SN) whose size, variety, or update rate exceeds the capacity of commonly-used spatial network computing to learn, manage, and process with reasonable effort [8]. The use of BSN is rapidly expanding into areas like mobility, route planning, navigation, flow analysis, etc. When a BSN is too big to be managed, the design of efficient storage methods in cloud computing environments are needed. Moreover, new scalable query processing algorithms over BSN should be investigated to answer different kind of queries (i.e. graph-oriented, location-oriented, trajectory-oriented, etc.). Therefore, the main target of the present research proposal is to overcome these new and exciting challenges.

Nowadays, the volume of available spatial data (e.g. location, routing, navigation, etc.) is increasing fast all over the world. Recent developments on Big Data have motivated the emergence of novel technologies for processing large-scale spatial data on clusters of computers in a distributed fashion. These *Distributed Spatial Data Management Systems* (DSDMSs) can be classified in disk-based and memory-based ones. The disk-based DSDMSs are characterized by being Hadoop-based systems (e.g. Giraph [1], SpatialHadoop, etc.) and they enable to store big spatial datasets and execute spatial queries using predefined high-level spatial operators without having to worry about fault tolerance and computation distribution. The memory-based DSDMSs are characterized as Spark-based systems (e.g. GraphX [7, 2], SpatialSpark, etc.) and they allow users to work on in-memory distributed data, without (like in Hadoop-based systems) worrying about the data distribution mechanism and fault-tolerance.

In this paper, we present our emerging work towards designing new storage methods and query processing algorithms over BSN in scalable and distributed systems. Section 2 contextualizes the starting points of our research. Finally, Section 3 explains our motivation and the objectives of the intended research.

2 Context of Research

Motivated by the increased interest to store, manage and analyze large volumes of data; new models, methods and algorithms have to be proposed to scale and perform well in distributed environments. Therefore, scalable and efficient storage and query frameworks are critical to realize the full value of BSN.

2.1 Storage Methods for Big Spatial Networks

Traditionally, *networks* are modeled as a graph *G(V, E)*, where V denotes the set of vertices and E denotes the set of edges of the network. Of particular interest

is a weighted graph, where a weight is associated with each edge. A *spatial network* is an extension of a network such that additional spatial components are associated with the elements (vertices and/or edges) of the graph.

Commonly, *spatial network partitioning* refers to the technique of partitioning a spatial network into multiple disjoint sub-networks or partitions in such a way that the partitions have homogeneous properties inside them. The spatial network partitioning can be considered as a *graph partitioning* problem, and for this reason, we can find many approaches in the literature. Recently, *multi-level graph partitioning* approaches have been proposed, and one of them is the *G-tree* [9]. It is inspired by the classical R-tree on Euclidean space, and it has two important characteristics. First, it is a balanced-tree structure, since the spatial network is constructed by recursively partitioning the original spatial network into sub-networks and each G-tree node corresponds to a sub-network. Second, the G-tree enables best-first search on such tree-based index, since the best-first algorithm has been widely applied in many spatial queries, showing an excellent performance. With this advantages in mind, new variants of the G-tree have been proposed to answer queries related to the graph structure of the spatial network and based on distances. To organize and store BSN in scalable and distributed environments, this new index structure can be used, as in a similar way that R-tree is used in SpatialHadoop or SpatialSpark to manage and store large spatial datasets.

2.2 Query Processing over Big Spatial Networks

Query processing on spatial networks differs from query processing on spatial databases, since the objects on a SN are restricted to move on the predefined paths of the underlying network. This has the important outcome that the path between two spatial objects depends on the actual connectivity of the network, rather than the relative distance between the objects. Consequently, the algorithms for processing the SN counterparts of the common spatial queries such as range and nearest neighbors have to consider the connectivity of the network in order to provide a correct answer to the queries. According to [4], the queries over SNs can be classified as follows:

1. **Graph-structure-oriented queries**. Shortest path, Minimun spanning tree, Route planning, Hamiltonian path, etc.
2. **Spatial-oriented queries**. Nearest neighbor queries, Distance-based joins queries, Aggregation queries, Spatial keywords queries, etc.
3. **Mobility-oriented queries**. Continuous nearest neighbor queries (CNNQ), Sequenced route queries, Trajectory-oriented queries, etc.
4. **Other SN-based queries**. Uncertainty queries, Queries with obstacles, etc.

Nowadays, there are several *distributed graph-parallel systems* based on Hadoop (e.g. Giraph) or Spark (e.g. GraphX), but the incorporation of queries over BSN in these complex systems has not attracted the attention to the research community. For this reason, this is an exciting research challenge, since current

methods, models and algorithms do not always scale and/or perform well when storing, managing, and analyzing large volumes of data in BSN.

3 Motivation and Objectives

According to [8], the development of *Spatial Network Big Database Systems* (SNBDS) requires overcoming three key challenges. First, it requires new data models to represent the complex and interrelated structure of SNBD. Second, fully exploiting SNBD requires scalable query processing and optimization methods. And third, SNBDS demands efficient I/O storage and access methods that leverage scalability and efficiency of large datasets. Motivated by these three challenges, we will focus our research in the design of new storage methods and query processing algorithms over BSN in scalable and distributed systems. In particular, our research objectives are the following:

1. Experiment with distributed graph-parallel systems based on Hadoop or Spark to store BSN from Open Street Map (OSM).
2. Provide to distributed graph-parallel systems of multi-level graph partitioning techniques to store BSN efficiently.
3. Design new algorithms for spatial-oriented queries (e.g. distance-based join queries) and mobility-oriented queries (e.g. CNNQ and trajectory-oriented queries) over BSN with mobility data in distributed graph-parallel systems.
4. Since main-memory resources are critical in Spark-based systems, we can consider the possibility to compress the mobility data (e.g. trajectories [3]).
5. Execute experiments with the new proposals (storage methods and query processing algorithms) over BSN in distributed graph-parallel systems, analyzing, comparing and drawing conclusions from the results.

References

1. Apache Giraph, `http://giraph.apache.org/`, 2013.
2. Apache Spark GraphX, `https://spark.apache.org/graphx/`, 2014.
3. N.R. Brisaboa, T. Gagie, A. Gómez-Brandón, G. Navarro, J.R. Paramá: "Efficient Compression and Indexing of Trajectories", *SPIRE Conference*, pp. 103-115, 2017.
4. N. Pelekis, Y. Theodoridis: "Mobility Data Management and Exploration", *Springer*, 2014.
5. S. Shekhar, S.K. Feiner, W.G. Aref: "Spatial Computing", *Commun. ACM* 59(1): 72-81, 2016.
6. S. Shekhar, V. Gunturi, M.R. Evans, K. Yang: "Spatial Big-Data Challenges Intersecting Mobility and Cloud Computing", *MobiDE Conference*, pp. 1-16, 2012.
7. R.S. Xin, J.E. Gonzalez, M.J. Franklin, I. Stoica: "GraphX: A Resilient Distributed Graph System on Spark", *GRADES Conference*, pp. 2, 2013.
8. K. Yang, S. Shekhar: "Spatial Network Big Databases - Queries and Storage Methods", *Springer*, 2017.
9. R. Zhong, G. Li, K.L. Tan, L. Zhou, Z. Gong: "G-Tree: An Efficient and Scalable Index for Spatial Search on Road Networks", *IEEE Trans. Knowl. Data Eng.* 27(8): 2175-2189, 2015.
10. A.Y. Zomaya, S. Sakr: "Handbook of Big Data Technologies", *Springer*, 2017.