Topological data structures for spatial analysis on Hadoop

01 Introduction

01. Geospatial topology

Determining topological relations among spatial objects are frequently used operations both in Geographic Information Systems (GIS) and computer graphic applications. Widespread implementations are usually based on the *Dimensionally Extended Nine-Intersection Model*. This standard defines the spatial relationship of its two operands by the dimensional intersection of their interior, boundary and exterior region.

02. Topological models

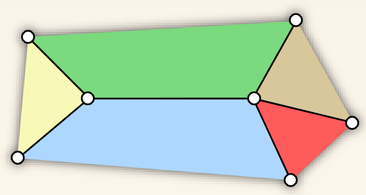
In many cases geospatial data sets are static or rarely altering, therefore the repeated recalculation of spatial relationships impacts the performance of geospatial operations needlessly. In these scenarios a topological representation created and cached in advance could significantly improve the computational efficiency of topological spatial queries.

Topological data structures like the *half-edge model* or the *winged-edge model* are widely used for the purpose of persisting the spatial relational information among the parts of the dataset, thus boosting the performance of later relational queries, meanwhile also eliminating the redundancy in the dataset.

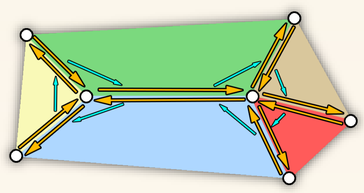
03. Distributed systems

Distributed systems following the *MapReduce* programming model like *Hadoop* have been around for years, proven to be an efficient framework for big data analysis, used in numerous fields including GIS researches. Distributed storage and processing of geospatial data sets are recently supported through various extensions (e.g. *SpatialHadoop* or *Hadoop-GIS)*, further easing the development of spatial data analytical applications.

While these systems provide complete functionality to execute topological spatial operations natively, they do not utilize the potential performance advantages of constructing and maintaining a topological representation of the spatial data sets.



Adjacent polygon faces.



Half-edge topological data structure.

02 Method

PREPARATION:

As an initial step the input vector geospatial data set is distributed between the processing nodes of the system. The data is chunked into *n* non-overlapping segments e.g. by the boundaries of a grid layout, thus creating smaller partial input files for the next phase.

Spatial objects overlapping multiple segments are split into them and are reunited in the last phase.

MAP:

Since the partial data sets are not overlapping, each part can be processed independently, resulting *n* topological sub-structures.

REDUCE:

In the final phase the partial topological data sets are merged with each other, producing a complete topological representation for the originally undivided input file. The merging process unites two subparts, therefore the reduce step can also be parallelized to increase execution performance.

The merging algorithm removes the duplicated vertices and edges on the segment boundaries, accordingly connecting the previously separated topological representations into a single graph. Unrequired edges and vertices created only for the partitioning of originally coherent but overlapping spatial objects are removed, therefore the output is not impaired by the segmentation of the initial input data set.

DISTRIBUTED TOPOLOGY CONSTRUCTION

03 Implementation

01. Topological representation

The processing of vector-based input data and the creation of the topological data structure was carried out with the intense usage and as part of the *AEGIS* geospatial framework.

The AEGIS system was initially developed for education and research goals at the *Eötvös Loránd University*, and is currently used both as a learning tool for computer science students and as a back-end engine for prototype implementations in GIS researches. It is based on well-known standards and state of the art programming methodologies and has been developed by taking adaptability and extensibility in mind. Utilizing the capabilities of AEGIS, like the support to handle a wide range of data formats (both vector and remotely sensed images) the implementation time and costs could be significantly reduced.

As a topological data structure, the half-edge model (also known as doubly connected edge list) was selected considering its support for fast traversal of the contained structure and its widespread usage in other GIS and computer graphics applications like the *Java Topology Suite* (JTS), *DotSpatial* or the *Computational Geometry Algorithms Library* (CGAL).

The implementation was carried out using the *.NET/Mono Frameworks*, because of the wide possibilities and the simple usage of this development platform.

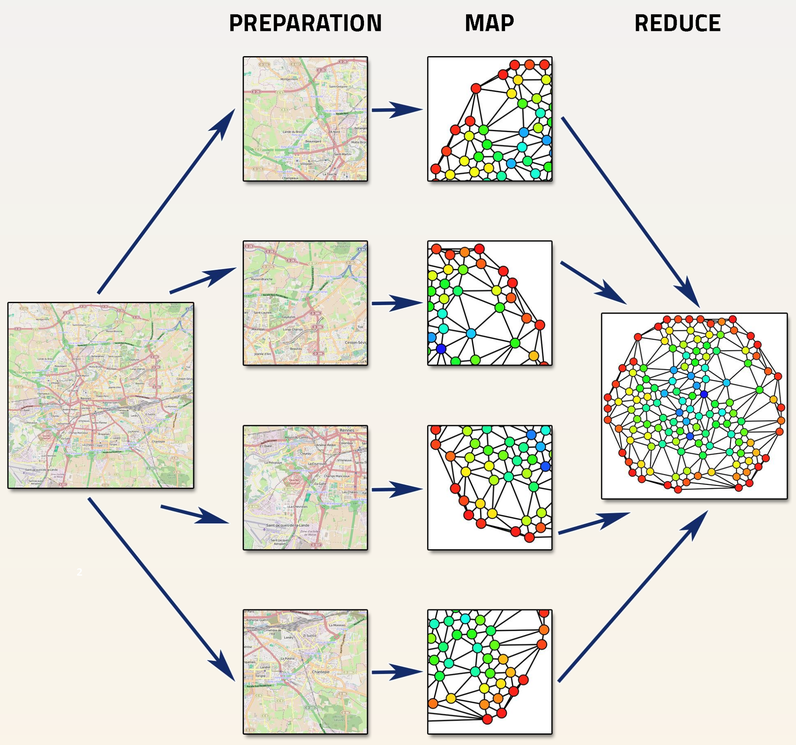


Image source: OpenStreetMap

The workflow of the applied MapReduce paradigm.

02. Distributed environment

The research to construct topological data structures in a distributed manner is part of the *IQmulus* (FP7-ICT-2011-318787) project. The main objective of the project is to enable (optimized) use of large, heterogeneous geospatial data sets for better decision making through a high-volume fusion and analysis information management platform.
Services in the project are based on the Apache Hadoop environment because of its proven outstanding reliability and scalability in large applications like *Yahoo* or *Facebook*. Therefore Hadoop was also used for the implementation purposes of this current research.

04 Conclusions

My research addresses the computational efficiency issues regarding topological calculations in large geographic information systems like the IQmulus project. Since repeated reproduction of the same topological information could considerably effect the execution time of spatial operations in these applications, topological data structures should be constructed and persisted directly after data upload or preferably at an (mostly) idle time of the whole distributed system. Through this approach later operations could be performed with multiple magnitudes of higher efficiency.

The topological analysis of large data sets and the creation of the topological representation for subsequent usage requires high computational capacity hence the available distributed system should be utilized to accelerate the progress. The research presents a general parallelization solution for building topological data structures through the MapReduce paradigm in a Hadoop environment.

It is notable that the demonstrated solution is not specific to Hadoop and also scales well with the number of available processing nodes, therefore it is not only applicable in distributed, multi-computer environments, but might also be used for optimization purposes in smaller systems.

Graphic Design EIT Digital Rennes - A.Cotais

Bibliography

1. C. Yang et al., Geospatial cyberinfrastructure: Past, present and future, *Computers, Environment and Urban Systems*, 2010.
2. A. Cary et al., Experiences on processing spatial data with MapReduce, *Proceedings of the 21st SSDBM*, 2009.
3. M. Mäntylä, An Introduction to Solid Modeling, *Computer Science Press*, 1987.
4. A. Eldawy, M. F. Mokbel, "A demonstration of SpatialHadoop: An efficient MapReduce framework for spatial data", *Proc. VLDB Endow.*, 2013.
5. A. Aji et al., "Hadoop GIS: A high performance spatial data warehousing system over MapReduce", *Proc. VLDB Endow.*, 2013.
6. R. Giachetta, A framework for processing large scale geospatial and remote sensing data in MapReduce environment, *Computers and Graphics*, 2015.

Information

Author: Máté Cserép
Contact: mcserep@inf.elte.hu
Date: October 19, 2015