Algorithms and Data Structures for Large Scale Geographic Information Systems: A Performance Analysis

Bernardo Marques
CISUC, DEI, University of Coimbra
Coimbra, Portugal
basimoes@student.dei.uc.pt

Luís Paquete
CISUC, DEI, University of Coimbra
Coimbra, Portugal
paquete@dei.uc.pt

Pedro Reino
Smartgeo Solutions
Lisboa, Portugal
pedro.reino@smartgeo.pt

Carlos Caçador
Smartgeo Solutions
Lisboa, Portugal
carlos.cacador@smartgeo.pt

Abstract

In this work, the performance of a system for the efficient processing of topological queries on geographic vector data, relying on in-memory algorithms and data structures, is analysed and compared with existing and available systems. The proposed system focuses mainly on querying time. Experimental results are reported and discussed in order to evaluate its performance.

Keywords: Geometric Data Structures, Spatial Indexing, Geographic Information System (GIS)

1 Introduction

This work aims to design and analyse algorithms and data structures for efficiently processing and storing geographic vector data.

This processing involves computing topological relationships between geometric data, such as touches, contains, overlaps and intersects predicates, as described in Open Geospatial Consortium's (OGC) ”Simple Features Specification for SQL”[2], based on the nine intersection model.

These data structures should not only allow for the storage of geographic data in memory, synchronized with an existing relational database, but also the performance of efficient spatial queries. The algorithms for performing these queries are analysed experimentally and compared against postGIS spatial extension, using data provided by Open Street Maps (OSM) for that purpose.

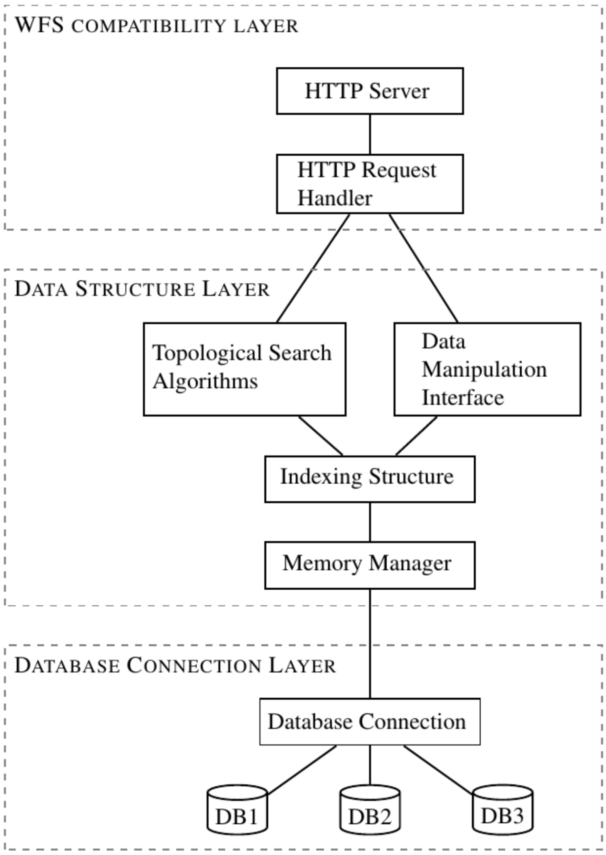
In order to improve the efficiency of query operations, the present system assumes that the data structures will live mostly or completely in-memory.

2 Architecture

In order to devise a solution that complies with the requirements presented in the previous section, a modular architecture was designed as depicted in Figure 1.

The system is divided into three layers, each comprised by a collection of modules. Within the scope of this project lies the ”WFS Compatibility” and the ”Data Structure” layers. The ”WFS Compatibility Layer” expects requests compliant with OGC's Web Feature Service Interface Standard[4], which are then parsed by the “Request Handler” and solved using functions from the ”Data Structure Layer”. The ”Data Structure Layer”, provides an interface for searching and managing data using an in-memory spatial indexing structure.

Figure 1: Top-level system architecture.



The “Memory Manager” module is used to prevent excessive memory consumption by the indexing structure, thus being capable of loading and saving the structure to external memory.

Figure 2: Which polygons cover point x?

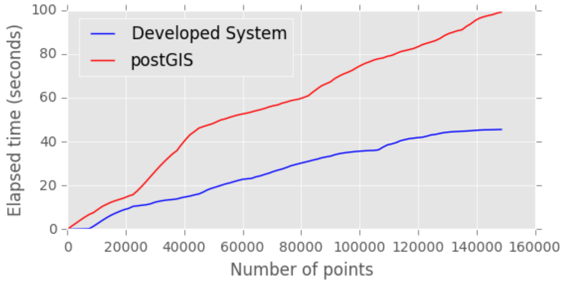
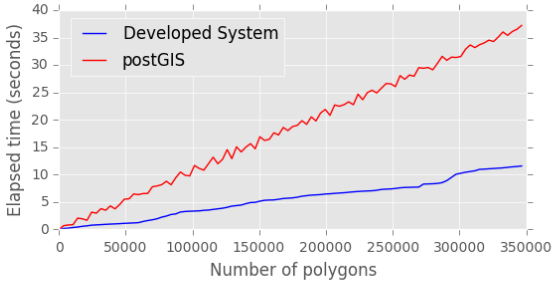


Figure 3:Which points are covered by polygon x?



3 Data Structure

Since the main concern is query performance, the chosen data structure for spatial indexation is the Polygonal Map Quadtree [3]. Since this structure is based on partitioning the space into disjoint cells, it is expected to perform better than a non-disjoint partitioning structure [1], even though it has probably higher memory requirements. Moreover, relatively good update times are also an important factor in this choice.

In order to process the topological queries, the data structure is used as follows:

1. for each node intersected by the query geometry, update a 9-intersection matrix for each geometry present in the node.
2. For each node that is strictly contained in the query geometry, iterate over these node's geometries and update the respective 9-intersection matrices.
3. Starting at one point of the query geometry, iterate over all nodes intersected by a line starting at that point and ending outside the tree domain. Any polygon that completely contains the query geometry and has not been analyzed yet, is stored in one these nodes.

Using the above description, it is possible to build a 9-intersection matrix for each relevant geometry, which results in the ability to answer most topological predicates with similar times. Moreover it is also possible to report more than one topological predicate without much additional effort.

4 Experimental Results

In order to compare the performance of the presented system against postGIS, data from Open Street Maps regarding the geographic region of Portugal was used.

These results were computed in a laptop with an Intel Pentium P6000 (1.87GHz, dual-core), 4GB DDR3 SDRAM and a solid state drive “Samsung 850 EVO”, connected using SATA II technology. The operating system is a minimal Arch Linux instalation without graphical environment. The source code is written in C++ and compiled using GNU Compiler Collection.

The results for two queries are presented:

1. In Figure 2, the results for the query “Which polygons cover point x?”
2. In Figure 3, the results for the query “Which points are covered by polygon x?”

The plots indicate that the performance of the presented system is slightly better than that of postGIS. It is also worth noting that the results for postGIS consider only the execution time as reported by the *explain analyze* SQL command.

Note that the proposed system can still be improved, for instance, by using more accurate aproximations to geometries to avoid unnecessary and costly calls for the computation of intersections between geometric objects, which is now done using minimum bounding boxes.

References

[1] Erik G. Hoel and H.Samet. A qualitative comparison study of data structures for large line segment databases. *ACM SIGMOD Record*. Vol. 21. No. 2. ACM, 1992.

[2] Open Gis Consortium Inc. Opengis simple features specification for sql. *Technical Report Revision* 1, 1999.

[3] H. Samet and R. E. Webber. Storing a collection of polygons using quadtrees. *ACM Transactions on Graphics (TOG)*, 4(3):182–222, 1985.

[4] P. A. Vretanos. Web Feature Service 2.0 Interface Standard. *Open Geospatial Consortium*, 2010.

Acknowledgements

This work is being developed in the context of the R&D project QREN unit in co-promotion no.34164 - Smartgeo Portal ”Geographic Information Management System”. The project is co-financed by QREN and Innovation Agency within the scope of the technological development and research incentive system, by the Operational Program for Competitiveness and Internationalizations (Compete).