Improving Geospatial Semantic Web Query using Optimization techniques

Chuanrong Zhang[a], Tian Zhao[b], Weidong Li[a]

[a]Department of Geography & Center for Environmental Sciences and Engineering, University of Connecticut, Storrs, CT, USA.
[b]Department of Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI, USA.

Introduction: The SPARQL was developed for Querying RDF data on Geospatial Semantic Web. However, SPAQL query may become very slow when query a large number of spatial objects. This study investigated several optimization techniques used for improving performance of SPARQL query. We focused on the client side optimization techniques in this study.

Client-based solution has several advantages over alternative approaches to address the semantic gap and performance problems. One is that it does not require a centralized server to coordinate query processing. The query workload is more evenly distributed between clients and servers, where the clients handle the spatial joins of data from different sources. The second advantage is that data sources provide efficient query performance through their native data storage system that can take advantage of existing spatial indices. There is no storage overhead of ontology-based encoding since ontology-based queries are translated to WFS queries. The third advantage is that client-side optimization helps reduce network traffic by reusing results from previous queries, which are not possible with a server-centric solution. The major investigated techniques include on-the-fly spatial indexing, tile-based rendering, query rewriting algorithms, efficient algorithms for spatial join and cache techniques (Figure 1).

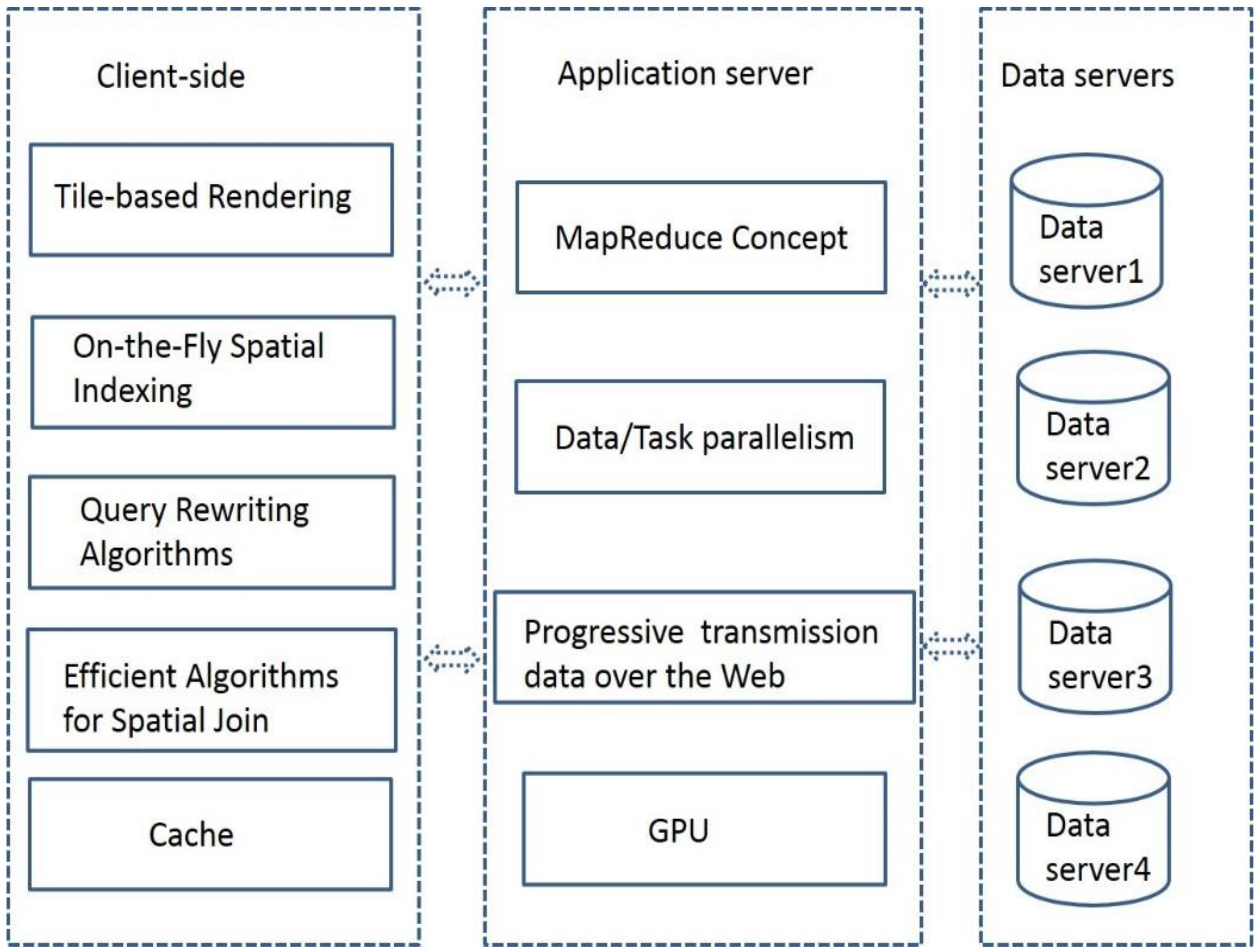

Fig. 1. The major techniques used for improving the performance.

A prototype implementation: In this case study, we aim to develop a prototype for flood disaster response based on Geospatial Semantic Web technologies for New Haven, a coastal city on Long Island Sound in Connecticut, USA. We expect the developed prototype will facilitate a fast flood response via better sharing updated data/information among municipalities and electric utility service providers.

The implemented prototype is accessible from the website: http://tianpar.cs.uwm.edu:8080/nh-hydro/. Geoserver (version 2.13.0) (http://geoserver.org/), an open-source software system, is used to publish spatial data in OGC WFS and WMS. Figure 2 shows the interface for the implemented prototype.

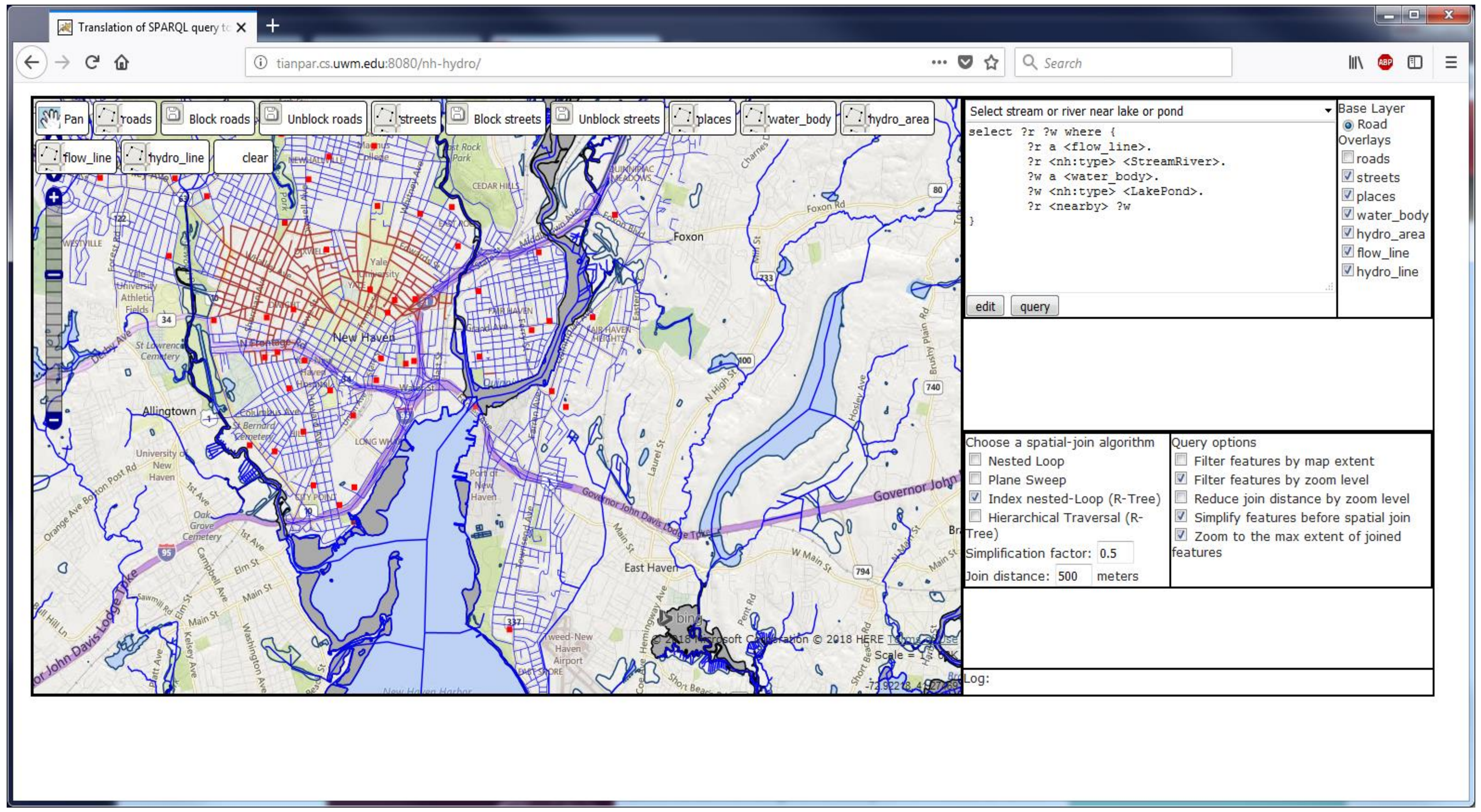

Fig. 2. Interface for the web application of the implemented prototype.

Table 1 lists how long it takes to conduct the SPARQL queries using different strategies.

Query examples	A	B	B + C	B + D	B + C + D
Select flow line near water body (4995 polygons and 10654 lines)	73.307	6.342	4. 997	1. 818	0.785
Select stream or river near lake or pond (2522 polygons and 4818 lines)	23. 633	2.514	0.395	0.297	0.028
Select streets near high schools (9 points and 3449 lines)	2.234	2.200	2.172	0.155	0.073
Select streets near "Fair Haven School" and less than 500 feet (1 point; 2481 lines)	1.786	1.690	1.760	0.019	0.016
Select roads near middle schools (4 points and 1537 lines)	0.879	0.871	0.874	0.036	0.029
Select high schools near state highways (9 points and 29 lines)	0.217	0.211	0.226	0.018	0.006

Table 1. Performance of spatial join query examples (unit: seconds). (A – nested-loop algorithm; B – R-Tree index nested-loop; C – Tiled-based rendering; D – Cache)
.

Conclusion: Our experimental results show that the Geospatial Semantic Web query suffered from the slow performance without using these optimization techniques. The optimization techniques such as on-the-fly spatial index, tile-based rendering, and cache techniques can greatly improve the performance of SPARQL query. The spatial join queries using different spatial join algorithms such as nested-loop algorithm, R-Tree index nested-loop algorithm, and tiled-based rendering algorithm have different performance. Although these optimization techniques can improve the performance of the prototype, there are still some limitations. Other optimization techniques such as cloud and parallel computing, or GPU techniques should be adopted in the future to further improve the Geospatial Semantic Web query.