12-9-2018

Scalable Spatial Framework for NoSQL Databases - Haslam Scholars Program Undergraduate Thesis

Daniel F. Enciso
denciso@vols.utk.edu

Scalable Spatial Framework for NoSQL Databases

Haslam Scholars Program
Undergraduate Thesis

Daniel Enciso

Haslam Scholars Cohort 2014

Expected Graduation December 2018

Submitted December 9, 2018

All work for this thesis was completed by the end of Spring 2017 semester.

Table of Contents

Abstract

The spatial frameworks used for knowledge discovery in “Big Data” areas such as urban information systems (UIS) are well- developed in SQL databases but are not as extensive within certain NoSQL databases. The focus of this project is to develop this framework for emerging search systems (ESS) in UIS by utilizing NoSQL databases, notably the document-based MongoDB. Such framework includes spatial functions for the most fundamental spatial queries. An ESS in UIS can take advantage of these new and attractive features of scalability within MongoDB to provide a robust approach to spatial search that differs from SQL relations and scalability. MongoDB, which is relatively in its early stages of spatial search in contrast to PostgreSQL, will require contributions to its spatial “toolbox”. Many of the operations present in SQL packages, such as PostGIS, are not in MongoDB. Thus, there is an opportunity to contribute to MongoDB’s ongoing geospatial evolution by developing, testing, and optimizing the spatial utilities used for large NoSQL datasets. Within UIS, these core operations can prove to be an important starting point for detailed geospatial analysis and high-impact data production. We hope, by open sourcing this framework (as an extension), it can serve the research community as the foundation for scalable NoSQL platforms for big geospatial data analytics and be the next stage for open source contributions to MongoDB.

About Me

Name:	Daniel Enciso
Institution:	University of Tennessee, Knoxville
Major:	Computer Engineering
Years Attended:	2014 – 2018
Department:	Min Kao Electrical Engineering and Computer Science
College:	Tickle College of Engineering
Scholars Programs:	Haslam Scholars Program (HSP) Min Kao Scholars Morris Scholars
Research:	Applied Software Engineering
Location:	Oak Ridge National Laboratory
Mentor:	Rajasekar Karthik
Faculty Advisor:	Arvind Ramanathan
HSP Contact:	Sylvia Turner
Hometown:	Franklin, TN

More information can be found on my LinkedIn page: https://www.linkedin.com/in/danenciso/
For references, please contact me at denciso@vols.utk.edu and I can put you in touch with them.

About this Thesis

From the very beginning of my time at the University of Tennessee, Knoxville (UTK), there has always been a strong encouragement from faculty and staff to get involved with research. As I voiced my interest in studying computer software and hardware, certain individuals would reach out and indicate the possibility of getting involved with Oak Ridge National Laboratory (ORNL). This definitely caught my attention and it seemed like a serious step in the right direction. Given that there are opportunities to work in labs at the university, I believe the unique aspect about being a student at UTK is to somehow become involved at ORNL. So, I preferred the latter over the former. It has been an incredibly insightful time and I can honestly say this era in my life is undoubtedly the catalyst for my future efforts.

Although my entire life journey has brought me to the level I am at currently, my journey with respect to research has played a pivotal role for my profession. My goal with this paper is to describe this journey in a way you, the reader, may understand this narrative in both technical and non-technical terms. The substance will focus on my development as a researcher and professional. However, I also seek to share the "how-to" aspects of my path. I believe this is equally as important as this reveals my perspective of my work and the appreciation I have for the individuals and organizations that were a part of this journey.

The rest of this paper, after some acknowledgements, will build up the details for the thesis and then ultimately explore the core of the work. It will conclude with some of the work that I have engaged in after ORNL and some possible next steps in terms of graduate work.

Acknowledgements

Without a doubt my efforts determine the results I obtain. Yet, I know the immense value it is to have a network that opens your mind, encourages to pursue your dreams, and connects you with the right people and opportunities. I would never have developed the vision I have for my life without the interactions I have enjoyed and learned from during my lifetime.
I would like to thank the individuals who had a major role in my involvement at ORNL and subsequently on my thesis experience. HSP and ORNL joined efforts to get scholars interested in

undergraduate research; I was one of these students. I want to thank Nicole Fazio-Veigel, who previously led the Office of National Scholarships & Fellowships at the University of Tennessee. Through her course and our conversations, she connected me with Dr. Taylor Eighmy, the Vice Chancellor for Research and Engagement at the time. My meetings with Dr. Eighmy were essential to understand the value of research and the path that can result from an experience at ORNL. I also want to recognize the role of Dr. Shaun Gleason, Director of the Computational Sciences & Engineering Division, and Dr. Budhendra "Budhu" Bhaduri, Group Leader of Geographic Information Science and Technology (GIST). They believed in my potential and took the chance of setting me up with a summer internship. Karthik Rajasekar has been my long-time mentor at ORNL; without his advice and determination to challenge me I would not have the self-learning skills and mindset needed to succeed. Dr. Arvind Ramanathan teamed up with Karthik to help me with

To everyone else in the GIST group and the HSP staff, thank you for sharing your perspectives and experience. To my fellow interns over the years and fellow scholars, the balance and diversity permitted me to understand a range of life paths. To advisors in my college including Travis Griffin and Dr. Leon Tolbert, thank you. My professors prepared for the challenges of research through my exposure to the fundamental knowledge needed to work creatively and consistently.

My time at UT would not be possible without the financial contributions via scholarships provided by the state of Tennessee, UTK, and generous donors/believers like Dr. Min Kao and Steve & Laura Morris.

Finally, I want to thank my friends for their support as well as my siblings, Sergio & Laura, and my extended family for their impact on my development… permitting me to become wise beyond my years. Additionally, without the efforts of my parents, Wilson & Nidia Enciso, I would not be nearly anywhere close to where I am today.

Research Experiences

My involvement with research at ORNL began in 2015 and ended in 2017. Every single step has been an equally important. Each stage added a next level of skills. They are described below.

Summer 2015

Summary

This was a full-time internship. I had just completed my freshman year at UTK and joined the GIST group for a unique learning experience. One undeniable lesson that I learned from my time there was growth through self-learning. My mentor, Karthik, provided the main goals and overview of my time there. Although quite simple in hindsight, these tasks and the level of quality expected from me were, at the time, new challenges to me. In reality this was a time of accelerated growth to learn about software development, the professional workplace environment, national lab organization, and working with colleagues at different levels of education.

While still too young to take on a project of my own, I was able to contribute to an existing project that Karthik had invested much effort and creativity to by the point I joined the team. Our regular meetings helped me stay on track and, more importantly, understand the larger scope of the project. Karthik taught me how research works at a national lab, from proposals and funding to results and knowledge sharing. It really was a comprehensive experience and made a significant difference on the technical skills I learned (e.g. front-end programming) and professional skills (e.g. workplace conduct and idea communication).

I presented my research via a poster at the end of the summer and I was also able to go to the Tennessee State Capitol to share my research there. I was able to prove that I was aware of what I had worked on by presenting my work and this would put me a step ahead in the public speaking skills necessary to effectively and clearly communicate ideas and results.

It was a successful summer and Karthik made it clear that I could continue working if I wanted to work on future projects in the GIST group at ORNL.

Abstract

Developing Rich and Interactive User Interfaces for the Analysis of Strategic Materials

Daniel Enciso (University of Tennessee, Knoxville, TN 37966) Rajasekar Karthik (Oak Ridge National Laboratory, Oak Ridge, TN 37830)

In this volatile global economy, securing the supply of strategic and critical materials plays a major role in the national security interests of the United States. Supply chain decomposition requires

locating and researching the mines, facilities, and companies associated with the production of that material. Analysts face two major challenges in decomposition: (1) vast quantity of data (i.e. "Big Data") and (2) constantly changing supply chains. The analyst utilizes a multitude of sources. One such source is the Dun & Bradstreet (D&B) company—D&B provides the world's largest commercial database with over 240 million company records and updates 5 million times per day. Other sources used are products information, mining news, events, financial information, etc. for the decomposition of one supply chain. News organizations also provide information on hundreds of articles on any single topic every day. The visual presentation and physical accessibility of this knowledge becomes crucial in order to identify information with new insights. Strategic Materials Analysis & Reporting Topography (SMART) is an analytic information system developed at ORNL to provide situational awareness of strategic material production and supply chain as well as supporting analysis of potential future outcomes. In this project I developed various components: (1) rich and interactive user interface for easy and fast visualization of the above information, (2) database schema for data storage, and (3) helper utilities such as extract -transformation-load (ETL) and feed aggregation processes. This project was completed successfully and will be integrated into the SMART software. The end use of these efforts will assist the decision makers significantly as the analyst will quickly discover leads and information with high impact factors.

Spring 2016

Summary

At this point in my undergraduate career, I had exposure to the "weed-out" classes in my major which included Data Structures & Algorithms I and Circuits I. I had just completed the toughest schedule with a total of 19 hours in the Fall of 2015. Towards the end of that semester I had contacted Karthik to see if I could join a project for the Spring 2016 semester.

This role was part-time and proved to be a new challenge because of the course work I still had to complete and the commute I had to take to be present at the lab. I believe I reached another level of maturity in my professional career. Excelling in this new environment was still my goal but the approach had to be well-managed so that my studies and work would receive the same amount of attention. Additionally, with this new experience Karthik insisted that I learn something new so that is why I transitioned into the back-end programming aspects. While this still remained a part of the same project I had already collaborated on, I knew exposure to new tools would serve me well. Once again, I presented this research, and this was another positive outcome of my time spent at ORNL. Not only was I developing knowledge and skills in the area of applied software engineering research but also fully acquiring the well-rounded capabilities to be a leader on projects in both academic and professional settings. While still only a sophomore at that point, I knew that ORNL and my efforts there were changing me for the better.

Through the completion of my work that semester and having in mind the work I had completed that summer, I decided that it would be best to work on an undergraduate thesis with the same team. I voiced these thoughts with Karthik and we both worked to make this a possibility.

Abstract

Fundamentals of Enterprise Applications: Server -side and Middleware Development

Daniel Enciso (University of Tennessee, Knoxville, TN 37966) Rajasekar Karthik (Oak Ridge National Laboratory, Oak Ridge, TN 37830)

Oak Ridge National Laboratory provides technical support for securing strategic and critical materials during national emergencies. This support is provided through a computational system called Strategic Materials Analysis & Reporting Topography (SMART). The goal of this project was to expand SMART's services by developing middleware tools for server-side operations. The SMART computational system is both a descriptive system and an analytical information system that will be used by analysts to decompose supply chains. An analyst considers multiple sources of data and faces constantly changing supply chains. SMART is being developed to mitigate these challenges and provide situational awareness of strategic material production and supply chain as well as supporting analysis of potential future outcomes. For our project we decided to program in Java, for middleware tools, and SQL, for database management. We implemented Java object oriented design patterns, specifically by encapsulating via (class) inheritance, abstraction within interfaces, and packages. We used these techniques to streamline the SMART web service through two components: (1) parsers and (2) data storage. We are experimenting now with an enterprise database to aid in supply chain decomposition and provide relevant results to specific queries. This project concentrated on correctly handling the update data in order to gain insights and provide tailored reports to analysts. We met our goal and successfully developed the parsing and storing tools to permit more SMART services for the future. This is important for two reasons: first, it provided an educational and professional experience with respect to enterprise (business) applications, and second, the advances we completed will allow analysts to receive the high-impact information they need, efficiently and effectively.

Fall 2016

This semester continued my experiences at ORNL while also serving as the beginning of the formal thesis investigation. An official course, HSP 497, was assigned for this work and the following description for the course was given by the instructor:

The course consists of a selection of Information Systems and Data Science topics for building scalable, high performance, and modular analytical systems. Topics may vary to reflect timely

research issues and the current interests of the instructor(s). Students are expected to complete a term project. Publications also may be required.

The literature review was necessary in preparation for a solid approach to the research project. In order to decide on a subject matter, wide range of sources were consulted. It will be apparent that the investigation involved obtaining information regarding spatial data structures and how such information can be accessed and stored. Further, there were some helpful resources for understanding the role of the NoSQL databases in geospatial applications. Some examples are provided below.

JOURNALS

Beckmann, Norbert, et al. "The R*-Tree: an Efficient and Robust Access Method for Points and Rectangles." *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data - SIGMOD '90*, 1990, doi:10.1145/93597.98741.

Fox, Anthony, et al. "Spatio-Temporal Indexing in Non-Relational Distributed Databases." *2013 IEEE International Conference on Big Data*, 2013, doi:10.1109/bigdata.2013.6691586.

Guttman, Antonin. "R-Trees." *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data - SIGMOD '84*, 1984, doi:10.1145/602259.602266.

Xiang, Longgang, et al. "Providing R-Tree Support For Mongodb." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B4, 2016, pp. 545–549., doi:10.5194/isprsarchives-xli-b4-545-2016.

After a literature review of a variety of resources (including research papers) and discussions with mentors, a completion plan was developed for this semester and the subsequent semester.

HSP 497 – HSP 498 COMPLETION PLAN

DANIEL ENCISO

FALL 2016 - 11/1/2016

OVERVIEW

- Prior to this project, two research experiences with Rajasekar Karthik at ORNL.
- Broad area interest: Data Science. Specifically, information retrieval using the best analytical tools (algorithms and indexing structures).
- Process to develop a specific research question has involved investigating where there is a potential to contribute new knowledge.
- Research Context: Emerging Search Systems provide benefits for Spatial Search, yet are missing components that well-established search systems provide.

PROPOSAL

- Original: Scalable Spatio-temporal Data Mining Techniques for Collaborative Filtering in Emerging Search Systems
- The details have been refined:
  - Developing geo-functions for an emerging search system
  - Improving collaboration and merge contributions when using existing and new geo-functions
- Ultimately, working on these two will provide scalable spatial search and improve the search experience

TASKS COMPLETED

- Beginning of Semester
  - 1) Select specific area of interest
  - 2) Read on existing search systems and differences (SQL vs NoSQL)
  - 3) Understand Information Retrieval
    - precision, recall, data association, collaborative-filtering, Bayesian networks
  - 4) Learn about Spatial Search
    - MBR, B-tree, R-tree, indexing, algorithms, modelling
- Middle of Semester
  - 5) Performance Analysis of SQL Queries using Different Indexing Structures
    - Shapefiles, geometry, spatial-relationships, best practices

TASKS REMAINING

- HSP 497
  - 1) Identify geo-functions of interest
    - Consult with GIST group
  - 2) Reverse Engineer geo-functions from traditional databases (SQL)
  - 3) Begin developing the equivalent functions for emerging databases (NoSQL)
- HSP 498
  - 4) Continue development of functions
  - 5) Test and improve
  - 6) Apply collaborative filtering and merge to improve search experience when using those specific geo-functions

TIMELINE AND DELIVERABLES

- Nov-Dec 2016
  - Tasks 1-3 from Tasks Remaining slide
- Jan-Mar 2017
  - Tasks 4-6 from Tasks Remaining slide
- Mar-May 2017
  - Finishing up any delayed items
  - Completing Poster/Oral Presentation/Research Paper

The semester concluded with development of a prototype of the tools to be developed in the final semester. The results are summarized below:

FINAL REPORT

Fall 2016 – HSP 497

Daniel Enciso

OUTLINE

- Overview
- Completion Plan
- ST_Area Development
- Discussion of Results
- Summary and Next Steps

OVERVIEW

- Final Completion of Semester Required:
  - Development of ST_Area for MongoDB
- Methodology:
  - Investigated ST_Area implementation in POSTGIS
  - Knowledge Migration to MongoDB
  - Adjusted function based on GeoJSON file (created from NYC shapefiles)
  - Confirmed Calculations MongoDB vs PostGIS

COMPLETION PLAN

- Refer to the other powerpoint.

- Important: Slide 5 (Tasks Remaining)
  - Tasks 1-3 were completed for ST_Area
  - A few more geo-functions will be added at the beginning of next semester
  - Will allow for more diverse spatial queries

ST_AREA DEVELOPMENT

- New York City Shapefiles
  - converted to GeoJSON
  - added to MongoDB database
- Analyzed two collections, neighborhoods and census_blocks
  - They contain geometry information for polygons
- PostGIS defines ST_Area:
  - Function that returns area of surface if it is a **polygon or multipolygon**
    - In this case, uses **SRID 26918 Projection Bounds** Values to calculate area
  - Reviewing GitHub source code for ST_Area
    - ST_Area uses **Shoelace Formula**
      - Finds the area of a simple polygon using Cartesian Coordinates

ST_AREA DEVELOPMENT

- Documents with collections contained:
  - Geometry information with coordinates (vertices needed for shoelace formula)
  - Needed to extract
  - Given in [Longitude , Latitude]
- Once stored in appropriate data structure and correct data type, coordinates sent to formula
- 1) Created .js file, test in mongo shell with load(file.js) command
- 2) Proceeded to implement the function as a mongoDB function
  - Loaded script to server
    - db.loadServerScripts();
  - Now simply call ST_Area(query);
    - Query must be:
      - ST_Area(db.census_blocks.find({"properties.BLKID":"360850009001000"},{"_id":0,"type":0,"properties":0}));
      - ST_Area(db.neighborhoods.find({"properties.NAME" : "West Village" }, {"_id":0,"type":0,"properties":0}));

DISCUSSION OF RESULTS

- ST_Area Function works for both collections
- MongoDB area output was compared with POSTGIS
  - Matched results with only a small difference in the 6th or 7th decimal place
- One issue to resolve is analyzing execution time for ST_Area in MongoDB
  - Need a simple way to see a time in ms

RESULTS

Results are in squared meters for ST_Area

Neighborhoods Data				
Query	West Village	East Village	Battery Park	Carnegie Hill
MongoDB	1044614.5296483561	1632116.7171849608	490191.7683069551	386517.51268965006
PostGIS	1044614.5296486	1632116.71718575	490191.768306941	386517.512689572

Census Blocks Data		
Query	360850009001000	360850020011000
MongoDB	22708.3516818434	10306.106295615435
PostGIS	22708.351681828	10306.1062956052

NEIGHBORHOODS COLLECTION/TABLE

MongoDB

```
> ST_Area(db.neighborhoods.find( {"properties.NAME" : "West Village" }, {"_id":0,"type":0,"properties":0},{"geometry" : 1 } ));
Area = 1044614.5296483561 square meters
> ST_Area(db.neighborhoods.find( {"properties.NAME" : "East Village" }, {"_id":0,"type":0,"properties":0},{"geometry" : 1 } ));
Area = 1632116.7171849608 square meters
> ST_Area(db.neighborhoods.find( {"properties.NAME" : "Battery Park" }, {"_id":0,"type":0,"properties":0},{"geometry" : 1 } ));
Area = 490191.7683069551 square meters
> ST_Area(db.neighborhoods.find( {"properties.NAME" : "Carnegie Hill" }, {"_id":0,"type":0,"properties":0},{"geometry" : 1 } ));
Area = 386517.51268965006 square meters
```

PostGIS

```
danielenciso=# SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name='West Village';
     st_area
-----------------
 1044614.5296486
(1 row)

danielenciso=# SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name='East Village';
     st_area
-----------------
 1632116.71718575
(1 row)

danielenciso=# SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name='Battery Park';
     st_area
-----------------
 490191.768306941
(1 row)

danielenciso=# SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name='Carnegie Hill';
     st_area
-----------------
 386517.512689572
(1 row)
```

CENSUS BLOCKS COLLECTION/TABLE

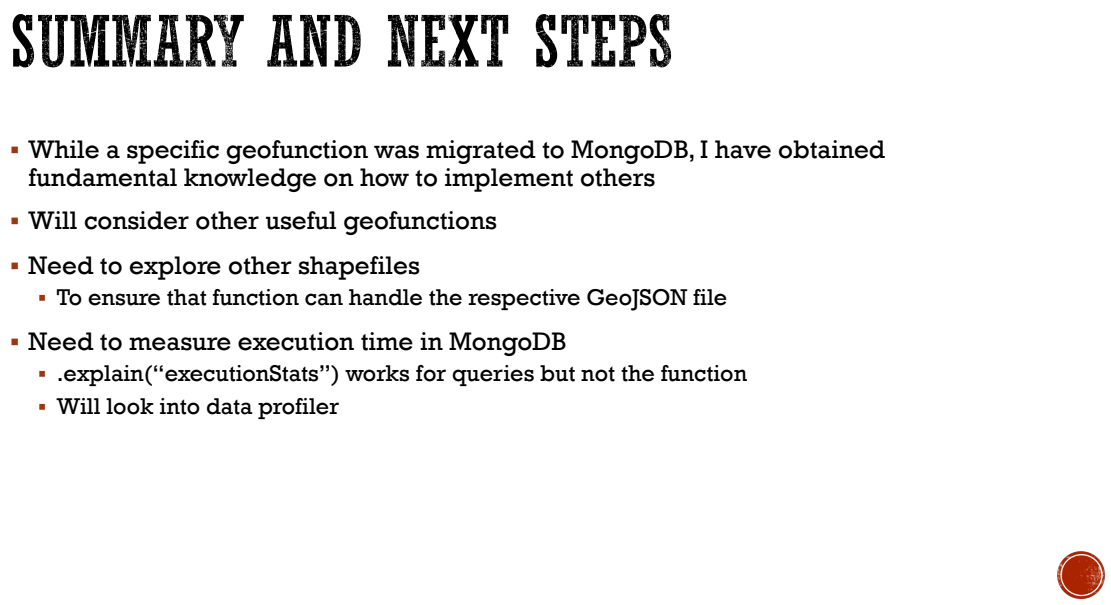
MongoDB

```
> ST_Area(db.census_blocks.find({"properties.BLKID":"360850009001000"},{"_id":0,"type":0,"properties":0}));
Area = 22708.3516818434 square meters
> ST_Area(db.census_blocks.find({"properties.BLKID":"360850020011000"},{"_id":0,"type":0,"properties":0}));
Area = 10306.106295615435 square meters
```

PostGIS

```
danielenciso=# SELECT ST_Area(geom) FROM nyc_census_blocks WHERE blkid='360850009001000';
     st_area
-----------------
 22708.351681828
(1 row)

danielenciso=# SELECT ST_Area(geom) FROM nyc_census_blocks WHERE blkid='360850020011000';
     st_area
-----------------
 10306.1062956052
(1 row)
```



Spring 2017

This semester focused heavily on the development of the tools (i.e. spatial functions) that could be used for spatial queries in the MongoDB database. Refer to demo paper for results.

Path to Thesis

Although briefly described previously above. There were specific steps taken to develop the exact research I wanted to pursue for the remainder of my time at ORNL. This all began with a conversation I had with my mentor Karthik towards the end of the Spring 2016 HERE experience. Since a great partnership of collaboration had already developed, I decided to continue working with the GIST. This would ensure I would progress to higher levels and, more importantly, the lab could take advantage of my consistent presence. Karthik recommended that we team up with Dr. Arvind Ramanathan to serve as a supervisor for our efforts. Dr. Ramanathan, a joint UT-ORNL faculty member, would provide the perspective of a professor and well-experienced researcher.

Demo Paper

The Demo Paper created for this thesis is included in the next pages. The format of the paper will show how it was prepared for the ACM SIGSPATIAL conference of that year (November 2017 at Redondo Beach, CA). However, it was not accepted by the conference so the information has been omitted so that there is no confusion. The demo paper was not presented at SIGSPATIAL.

Scalable Spatial Framework for NoSQL Databases (Demo Paper)

Daniel Enciso
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, TN, 37996
USA
denciso@vols.utk.edu

Rajasekar Karthik
Geographic Information Science & Technology
Oak Ridge National Laboratory
Oak Ridge, TN 37830
USA
karthikr@ornl.gov

Arvind Ramanathan
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, TN, 37996
USA
ramanathana@ornl.gov

ABSTRACT

The spatial frameworks used for knowledge discovery in "Big Data" areas such as urban information systems (UIS) are well-developed in SQL databases but are not as extensive within certain NoSQL databases. The focus of this project is to develop this framework for emerging search systems (ESS) in UIS by utilizing NoSQL databases, notably the document-based MongoDB. Such framework includes spatial functions for the most fundamental spatial queries. An ESS in UIS can take advantage of these new and attractive features of scalability within MongoDB to provide a robust approach to spatial search that differs from SQL relations and scalability. MongoDB, which is relatively in its early stages of spatial search in contrast to PostgreSQL, will require contributions to its spatial "toolbox". Many of the operations present in SQL packages, such as PostGIS, are not in MongoDB. Thus, there is an opportunity to contribute to MongoDB's ongoing geospatial evolution by developing, testing, and optimizing the spatial utilities used for large NoSQL datasets. Within UIS, these core operations can prove to be an important starting point for detailed geospatial analysis and high-impact data production. We hope, by open sourcing this framework (as an extension), it can serve the research community as the foundation for scalable NoSQL platforms for big geospatial data analytics and be the next stage for open source contributions to MongoDB. [1]

CCS CONCEPTS

• **Information Systems → Data Management Systems**; **Information Retrieval;**

GENERAL TERMS

Design, Performance, Scalability

KEYWORDS

Spatial Framework, Query Operations, Data Parsing, Geospatial Analysis, MongoDB, PostGIS, NoSQL

ACM Reference format:

1 INTRODUCTION

A spatial framework provides a geospatial analyst with the wide range of functions that can manipulate geometries stored in a database. These spatial frameworks were developed, as extensions for databases, to efficiently expand the management and analysis of spatial datasets. With the growth of these extensions over time (via the implementation of new functions and advances in the mathematical techniques used for spatial geometries), the queries of spatial search systems continue to provide the improved results required for high-impact reports. UIS are an example of area that requires a high-performance approach to knowledge discovery.

Such approach dictates that a scalable database, with a domain-specific (i.e. geospatial) extension, is available to serve as the backend structure for data management. The insight that is generated using the spatial functions will then influence major decisions in urban planning. Overtime, traditional (i.e. legacy) databases have adapted to the new data through the development of add-on extensions. Yet, these SQL databases have a standardized approach of storing and relating data, which dictates how the data can be queried. The distinct scalability approaches of SQL and NoSQL databases are important since they impact a search experience via advantages and disadvantages.

For UIS, there is the possibility to have significant gains in performance if NoSQL databases can reach the level of depth provided by the SQL geospatial extensions. Our project is interested advancing the progress of NoSQL-geospatial-based search systems to be high-performance applications for analysts. While SQL databases have been optimized for spatial search, that does not mean that they are the best storage and management choice for the data that will undergo geospatial analysis. It is beneficial to have more than one option.

Thus, we selected to develop a spatial framework for MongoDB. Section 2 discusses our study of spatial search and the chronology which lead us to make our claims. Section 3 gets into detail of the design of the framework and Section 4 demonstrates the use of this pilot framework and compares to an established SQL spatial extension. Section 5 concludes this demo paper, discusses ongoing development, and proposes future work.

2 SPATIAL SEARCH

2.1 System Specifications

We worked on one system, a MacBook Pro with macOS Sierra, 2.4GHz Intel Core i5, 8GB RAM 1600 MHz DDR3. The Terminal (Command Line Tools) was used for installing dependencies, vim for development. PostgreSQL, PostGIS, SQLite, and SpatiaLite were installed with Homebrew Package Manager (e.g. psql). MongoDB was installed and built from source from Git repository. JavaScript was our language of choice to interact with MongoDB. Thus, we utilized NodeJS, a JavaScript run-time environment for server-side execution, and npm, a package manager for JavaScript. MongoClient, performance and assert are a few of the packages that were included in development. For information visualization: QGIS, pgAdmin tool for PostgreSQL, Studio 3T IDE for MongoDB.

2.2 The Data

We used the same data set for the performance analysis and this demo. These shapefiles were obtained from an online tutorial which describes the number of records and table attributes for each dataset [1].

Census blocks data is composed of polygons, 36,592 records. It has demographic data and the following table attributes: blkid, popn_total, popn_white, popn_black, popn_nativ, popn_asian, popn_other, boroname, geom. Neighborhoods data includes 129 records (each a polygon spatial object). The attributes include name, boroname, and geom. Streets data contains a total of 19,091 records, i.e. linestrings. Its attributes are name, oneway, type, and geom. Fourth, the subway stations data consists of 491 records, i.e. points. The attributes: name, borough, routes, transfers, express, geom.

2.3 SQL Spatial Performance Analysis

In this section, we review the analysis completed by Performance Comparison of Spatial Indexing Structures for Different Query Types [2]. Pant gathered query execution time data for five categories of SQL queries and compared the influence of three indexes: R-tree (Rectangle tree), GiST (Generalized Search Tree), and R*-tree (Variant of Rectangle tree)[2]. The five categories: Simple SQL, Geometry, Spatial Relationships, Spatial Joins, Nearest Neighbors. He selected two database management systems and their respective spatial extenders: PostgreSQL with PostGIS and SQLite with SpatiaLite. He concluded his paper by stating R*-Trees provide shortest execution time in all categories excluding Simple SQL, R-Trees have the best execution time for Simple SQL queries, and GiST indexing can be considered for the Spatial Relationships, Spatial Joins, and Nearest Neighbors queries.

Query Category	Shortest Execution	Longest Execution
Simple SQL	GiST	R*-Tree
Geometry	GiST	R*-Tree
Spatial Relationships	R*-Tree	Without Index
Spatial Joins	GiST	Without Index
Nearest Neighbor	R*-Tree	Without Index

Table 1: Our Results from Performance Analysis Replication

After conducting our own performance analysis by following Pant's experiment, we came to a few conclusions. R-Tree indexing is no longer available for PostgreSQL [3]. In some instances, non-indexed queries were faster or as fast as GiST queries. GiST was superior in some cases (SimpleSQL, Geometry, Spatial Joins). R*-tree performance was best in Spatial Relationships and Nearest Neighbors, had similar time to GiST for Spatial Joins, and worst in SimpleSQL, and Geometry. Pant only included PostgreSQL queries, so we developed the SQLite queries (which may not be exactly the same as Pant's queries)

Ultimately, by understanding and testing the implementation of spatial indexing (e.g. GiST and R*-trees) and queries in SQL databases (e.g. PostGIS and SpatiaLite), we became familiar with traditional methods of spatial search.

2.4 NoSQL Spatial Database Motivation

The reasoning behind the use of a NoSQL database in spatial search is based off the inherent design of storing data non-relationally and serving as an efficient, distributed approach to data management. By removing itself from tabular relations, this provides an opportunity to be flexible when it comes to storing and searching the information.

For example, MongoDB uses JSON (JavaScript Object Notation) as it stores records with the basic data types (numbers, strings, Boolean values, arrays, and hashes) and represents these JSON documents in binary-encoded format (BSON) which provides more data types, ordered fields, and permits efficient database interactions within different languages [4].

2.5 MongoDB Spatial Search

MongoDB's geospatial query operations are, for the moment, few. This is not only evident by MongoDB's documentation but also Performance investigation of selected SQL and NoSQL databases [5]: huge amounts of data and frequent data changes characteristic of spatial search make a NoSQL database like MongoDB a great option. Given MongoDB's relative newness in the field of geo-information, their investigation is a demonstration of the shortcomings that are inherent in an evolving technology.

Function Call	Query Use Case
$geoWithin	Inclusion – locations entirely within
$geoIntersects	Intersection – locations intersect
$near	Proximity – query points nearest

Table 2: Current MongoDB Spatial Query Operations [6]

3 A PILOT GEOSPATIAL FRAMEWORK

3.1 Motivation

MongoDB's has potential within spatial search and this is possible through its data structure which has efficient cluster, performance, and data scaling [7]. By developing this

experimental spatial framework, it serves as a contribution to MongoDB and GIST efforts at ORNL. We suggest that the data management characteristics of NoSQL can improve spatial search. However, in the case of MongoDB, those features cannot be exploited without the spatial toolbox that matches the task (i.e. of "Big Data" UIS). Therefore, the functions we have developed and are developing will serve as a proof of concept.

3.2 Description

Foundational spatial functions were selected and ported from PostGIS to operate in MongoDB [8]. Refer to the following when reading the tables: Polygon = PO, Point = PT, LineString = LS.

	PostGIS	MongoDB
F1	**ST_BuildArea**	**st_buildarea**
Query Result: Aggregated PO from multiple PT or LS		
Use Case: Flexible method to identify and create area of analytical interest. Makes dataset dynamic and responsive.		
F2	**Box2D**	**box2d**
Query Result: Maximum-Cartesian-extents box of enclosed spatial objects (PT, LS, and/or PO)		
Use Case: Same premise as F1, yet distinct because it can help explain spatial relationships via operators		
F3	**ST_MakeLine**	**st_makeline**
Query Result: Aggregated LS from PTs and/or LSs		
Use Case: Like F1, this will enhance data manipulation through creation of new geometries (practical in designing cities, towns		
F4	**ST_Area**	**st_area**
Query Result: Measurement of spatial objects surface area		
Use Case: Computation delivering a quantitative aspect of a geometry is a basic need		
F5	**ST_Distance**	**st_distance**
Query Result: Measurement of 2-D Cartesian minimum distance between two spatial objects		
Use Case: Provides a solution to the common queries: How far away is point 1 to point 2?		
F6	**ST_DWithin**	**st_dwithin**
Query Result: Relationship between two spatial objects, true if within specified distance		
Use Case: Performs similar operation as F6, but is based on different demand: If people are only willing to travel a certain distance to the mall, will new development meet requirement?		
F7	**ST_Length**	**st_length**
Query Result: Measurement of 2-D of LS		
Use Case: Special case of F5, restricted to one spatial object: How far will/does this road extend?		

Table 3: PostGIS Port to MongoDB Geospatial Functions

Additionally, we propose other functions to pipe and filter relevant information within these functions (i.e. F1-F7). We decided to include these in the spatial framework because they provided further justification for our proof of concept: if the spatial functions can be built in a scalable form via reusability, then the contribution to MongoDB is notable.

F8	**main_BuildToArea**
Query Result: Pipe aggregated PO to area calculation	
F9	**main_BuildToDistance**
Query Result: Pipe aggregated POs to minimum distance calculation (MDC)	
F10	**main_ BuildToDWithin**
Query Result: Pipe aggregated POs and specified distance to MDC and comparison	
F11	**main_ LineToLength**
Query Result: Pipe aggregated LS to length calculation	
F12	**main_ LineToDistance**
Query Result: Pipe aggregated LSs to MDC	
F13	**main_ LineToDWithin**
Query Result: Pipe aggregated LSs and specified distance to MDC and comparison	
F14	**main_ Box2DToArea**
Query Result: Pipe maximum-Cartesian-extents box (MCEB) to area calculation	
F15	**main_ Box2DToDistance**
Query Result: Pipe MCEBs to MDC	
F16	**main_ Box2DToDWithin**
Query Result: Pipe MCEBs and specified distance to MDC and comparison	

Table 3: Proposed Piping Functions

Spatial Algorithms:
Shoelace formula is calculated with all the polygon's vertices, (x, y) pairs, to obtain the area of a non-intersecting shape (N sides).

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)$$

Figure 1: Shoelace Formula

Centroid of Polygon Formula uses the above formula for area A and the (x, y) vertices to locate the centroid of any non-intersecting polygon. Especially useful when polygons are irregular. Relevant to distance calculations between two polygons since distance is calculated between two points

$$C_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

Figure 2: Centroid of Polygon Formula

4 DEMONSTRATION

4.1 Using the Framework

Designing a scalable framework model was essential. The package is designed to add more functions. Section 3 described the individual spatial functions and as well as the piping functions to run combinations of spatial functions.

In each case, the goal was to have an editable .js executable file to serve as a script with command line instructions. This script contains the typical NodeJS standard for the Terminal: node runThisfile otherArguments. It contains the relevant query information for each command; this file calls the appropriate spatial function(s) and sends the arguments to the correct fields.

4.2 Individual Spatial Functions

The spatial functions described in Section 3 are straightforward in their purpose and use. GitHub has a well-documented repository for PostGIS, which assisted in acquiring a working knowledge of how the spatial functions work. Thus, having the reference was useful for porting the functions.

Example Query Approach of Area Calculation

PostGIS: SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name = 'Riverdale'

MongoDB: node st_area.js neighborhoods 0 Riverdale

Figure 3: Image of Verified Centroid (st_distance F5)

The calculation of the centroid was confirmed with: SELECT ST_AsText(ST_Centroid(geom)) FROM nyc_neighborhoods WHERE name='Bedford Park'; which returned POINT(594004.516130166 4525373.46926201). A number close to Figure 3: (594004.5161740588, 4525373.469594673).

4.3 "Piping" Spatial Functions

We proposed the piping functions in Table 4 to expand upon the functions we ported from PostGIS to MongoDB in Table 3. This was not done to compare performance between MongoDB and PostGIS, because PostGIS would have to have these piping functions develop (which is a viable option but not the aim of this demo paper).

Example Query Approach of Build To Area Operation

MongoDB: node main_BuildToArea.js ./st_buildarea ./st_area subway_stations 1 96th_St 21 0 103rd_St 22 0 116th_St 194 0 125th_St 41 0 110th_St 122 0 86th_St 119 0

4.4 Execution Time for Spatial Functions

This table displays the average query time execution (milliseconds) for functions F2-F7 described in Table 3. The All includes the first time (usually the longest time when the first query is sent). Thus the WF is without the first query time. An extra row under F2 and F4 exists because there were two query sets for those particular functions. The output matched for the queries in MongoDB and PostgreSQL. The differences in time are due to differences in how our scripts were set up. Further modification to the scripts are necessary to improve our results.

	PostgreSQL		**MongoDB**	
	All	WF	All	WF
F2	25.257	0.221	4.149	4.183
	9.135	2.559	3.023	2.891
F3	24.298	0.525	2.897	2.913
F4	40.145	0.178	0.858	0.515
	13.395	6.035	0.592	0.314
F5	4.461	0.344	3.831	4.058
F6	28.230	2.389	6.324	7.159
F7	11.053	2.585	0.715	0.379

Table 5: Numerical Comparison of PostGIS vs MongoDB

5 CONCLUSIONS

The goal of our framework is not only limited to providing the same functionality as PostGIS but also focused on: (1) simplifying the approach to spatial search via encapsulation of the functionalities and (2) having the relationships between functions (e.g. st_buildarea and st_area) already built in. We understand that our framework is one approach towards managing spatial queries and data in MongoDB. Providing this framework as open source will help us obtain feedback and improve this proof on concept.

Now that we have built some functions, the future continuation of this work will explore a large scale, performance analysis using MongoDB sharding and our framework. Section 2.1 shows that we did not test over distributed nodes. The motivation of our efforts (Section 3.1) can only reach the next level if we take advantage of horizontal scaling (dividing the system dataset and load over multiple servers) [10].

ACKNOWLEDGMENTS

This work was partially supported by the GIST Group at Oak Ridge National Laboratory. This work is part of the ongoing undergraduate thesis being completed by D. Enciso. I would like to thank my research advisors Dr. Arvind and Mr. Karthik.

REFERENCES

[1] Boundless. (n.d.). Introduction to PostGIS, 6. About our data. Retrieved May 02, 2017, from http://workshops.boundlessgeo.com/postgis-intro/about_data.html
[2] Pant, Neelabh. "Performance Comparison Of Spatial Indexing Structures For Different Query Types." (2015).
[3] Chapter 3. PostGIS Frequently Asked Questions. (n.d.). Retrieved June 21, 2017, from https://postgis.net/docs/PostGIS_FAQ.html#idm1080
[4] MongoDB. (n.d.). JSON and BSON. Retrieved May 02, 2017, from https://www.mongodb.com/json-and-bson
[5] Schmid, Stephan, Eszter Galicz, and Wolfgang Reinhardt. "Performance investigation of selected SQL and NoSQL databases." (2015).
[6] MongoDB. (n.d.). Geospatial Indexes and Queries. Retrieved May 02, 2017, from MongoDB Manual 3.4: https://docs.mongodb.com/manual/applications/geospatial-indexes/
[7] MongoDB. (n.d.). MongoDB at Scale. Retrieved May 02, 2017, from https://www.mongodb.com/mongodb-scale
[8] PostGIS. (n.d.). Chapter 14. PostGIS Special Functions Index. Retrieved May 02, 2017, from http://postgis.net/docs/PostGIS_Special_Functions_Index.html
[9] Bourke, P. (1997, July). Polygons and meshes. Retrieved May 02, 2017, from http://paulbourke.net/geometry/polygonmesh/
[10] MongoDB. (n.d.). Sharding. Retrieved May 02, 2017, from MongoDB Manual 3.4: https://docs.mongodb.com/manual/sharding/

Conclusion

My research experiences were fundamental to my college experience. The culmination of all those efforts into an applied research project in software engineering was definitely a challenging experience. More importantly, lessons were learned. Specifically, in the technical sense of acquiring the skills and knowledge of a researcher and software engineer. A major characteristic of my time in undergraduate research was the self-learning process in which my mentors constantly required me to experience the hardships of not knowing enough information and overcoming this obstacle by having the maturity and determination to become capable.

One suggestion on how the project could have been modified to enable better results: Instead of building functions outside of the MongoDB source code, actually develop them in there to take full advantage of the NoSQL innate abilities (e.g. horizontal scaling and spatial indexing). Due to my lack of software development knowledge at the time, I was not aware of the usefulness of taking source code that is available online (e.g. GitHub). This would have made the spatial functions much more scalable than just a JavaScript package that could be downloaded. Furthermore, the data was not tested over many nodes.

Originally, I could have decided to contribute to a current project at the lab for my undergraduate thesis. Instead, my mentor suggest and I accepted the opportunity to start something from scratch (relative to the research already completed). This investigation showed me the difficulties that come along with starting a project with limited support but that is still exciting because the research required some innovation. Fortunately, I was able to recognize the kind of thinking required for such innovation to occur through the guidance of my mentors (i.e. expert researchers).

Looking back, the project can definitely be improved, and my approach was not the best nor complete manner to develop the spatial functions. After completing the work, I was able to discuss with my mentor what could have gone differently. To me it became very clear that the only way one becomes a talented software engineer is through the learning process of building and breaking programs and systems. Having gone into college with no prior research experience, the learning curve was steep. Given the amount of time that I had I am glad to say that I did more than just complete some projects. I gained the ability to become a life-long learner who can analytically prepare solutions and always review, criticize, and improve the work that was done.