Research Statement

Jia Yu
jiayu2@asu.edu

Overview

The volume of geospatial data increased tremendously. Such data includes but is not limited to weather maps, socio-economic data, and geo-tagged social media. Making sense of the rich geospatial properties hidden in the data may greatly transform our society. Spatial data analysis is the technique that crunches massive spatial data to find insights. It includes many subjects undergoing intense study: (1) Socio-Economic Analysis: climate change analysis, study of deforestation, population migration, and variation in sea levels, (2) Urban Planning: assisting governments in city/regional planning, road network design, and transportation / traffic engineering. These data-intensive spatial analytics applications highly rely on the underlying data infrastructures such as database management systems (DBMS) to efficiently manipulate, retrieve and manage data. Unfortunately, classic database management systems suffer from a significant performance drop when handling large-scale spatial data.

Giving this context, my PhD thesis focuses on **crafting database systems to accelerate large-scale geospatial data analytics**. In particular, I worked on several challenging issues in this direction. First, I built a distributed cluster-computing system called GeoSpark to offer **scalable spatial data analytics** including spatial queries, visualization and traffic simulation. Second, I designed an approximate query processing system, namely Tabula, that sits between the data management system and the front-end visualization dashboard to uphold **interactive spatial visual analytics**. Finally, I designed a database indexing approach called Hippo to provide **fast yet lightweight index structures** (in terms of storage cost and maintenance speed). Besides, I collaborated with researchers at IBM Almaden Research Center and Microsoft Research in devising two machine learning based indexing mechanisms, Hermit and Alex.

My research outcomes have appeared in prestigious database / Geographic Information System (GIS) conferences and journals, including SIGMOD, VLDB, ICDE, Geoinformatica Journal, etc. All my research projects at school are open-source. I implemented GeoSpark ecosystem, Tabula system and Hippo index in the kernels of widely used open-source database management systems such as Apache Spark and PostgreSQL. My GeoSpark project has attracted users / contributors from major IT companies (e.g., Facebook, Uber and MoBike) and several startup companies. Databricks (the technical unicorn behind Apache Spark) provides a GeoSpark + Spark cloud environment. According to the source-code hosting websites (see statistics from Maven Central), GeoSpark ecosystem receives 10,000 downloads per month. I also helped my advisor to submit a proposal to NSF CISE IIS, titled as "III: Small: Towards Data Systems Support for Geospatial Visualization".

1 Current Research

1.1 Distributed spatial data analytics systems

Existing spatial data management systems, such as PostGIS and ArcGIS, extend relational DBMSs with new data types, operators, and index structures to handle spatial operations. Even though such systems provide full support for spatial data, they suffer from the scalability issue. The massive scale of available spatial data hinders traditional spatial query processing techniques from efficiently understanding the data. Researchers and practitioners recently extend Hadoop MapReduce, a distributed computation engine, to perform spatial analytics at scale. Although Hadoop-based approaches achieve high scalability, they still exhibit slow run time performance since they have to persist intermediate data on hard disks for every single operation.

Distributed in-memory spatial data management system. At Arizona State University, I investigated how to execute large-scale spatial queries at memory speed. Apache Spark, an emerging distributed in-memory cluster-computing engine based on Resilient Distributed Datasets (RDD), achieves an order of magnitude higher execution speed than the existing Hadoop MapReduce model. Unfortunately, Spark does not provide native support for spatial data and operations. Hence, users need to write tedious suboptimal programs to describe their own spatial data processing jobs on top of Spark. To remedy that, I designed and implemented GeoSpark [7], a distributed in-memory computing framework for large-scale geospatial data.

GeoSpark is an open-source full-fledged cluster computing framework that extends the core engine of Apache Spark and SparkSQL to support spatial data types, indexes, geometrical operations and spatial queries at scale. GeoSpark is equipped with an out-of-the-box Spatial Resilient Distributed Dataset. The Spatial RDD provides an API for Apache Spark programmers to easily develop their spatial analysis programs using operational (e.g., Java, Scala, Python and R) and declarative (i.e., Spatial SQL) languages. A research paper about GeoSpark was published in Geoinformatica Journal 2018 [7]. I also demonstrated GeoSpark in ICDE 2016 using several spatial analysis applications. In addition, I taught a tutorial on "Geospatial Data Management in Apache Spark" in ICDE 2019.

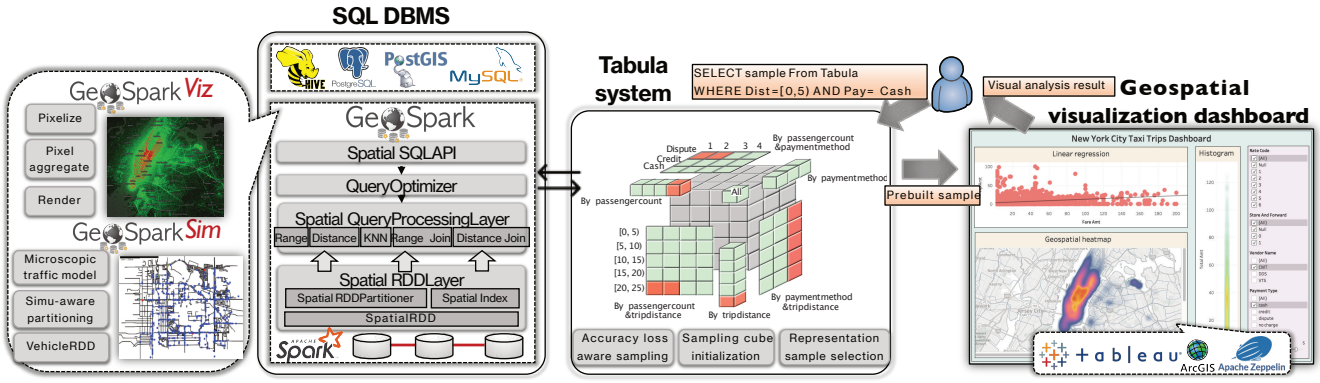

Fig. 1. An overview of large-scale geospatial data analysis. (1) GeoSpark ecosystem (datasystemslab.github.io/GeoSpark/) and other data engines serve as the underlying SQL DBMS. (2) Tabula system serves as the middleware (github.com/DataSystemsLab/Tabula). (3) Any spatial visualization dashboard can be the frontend.

Distributed map visualization system. The growth of GeoSpark offers me a unique opportunity to look into the spatial analysis cases from users. I realized that visualizing analysis results on maps may indeed assist researchers to better recognize the potential data patterns. However, the existing solutions simply pass massive spatial analytics results to lightweight visualization tools. Therefore, I designed GeoSparkViz [6] which combines both spatial analytics and visualization to deliver an end-to-end scalable visual analytics system. It provides native support for general cartographic design and seamlessly integrates with GeoSpark.

GeoSparkViz encapsulates the main steps of the geospatial map visualization process, e.g., pixelize spatial objects, aggregate pixels, into a set of massively parallelized RDD transformations in Apache Spark. Such RDD transformations provide out-of-the-box support for the user to generate a variety of high-resolution map visualization effects, e.g., scatter plot, heat map, and choropleth map, on Spatial RDDs. GeoSparkViz also proposes a map tile-aware data partitioning method that achieves load balancing for the map visualization workloads among all nodes in the cluster. Results of GeoSparkViz were published in SSDBM 2018 [6] and demonstrated in ICDE 2019.

Distributed road network traffic simulator. Road network traffic data contains the trajectories of a set of vehicles moving over a road network. Such traffic data has been widely studied in different disciplines that include urban planning, traffic prediction and spatial-temporal databases. Unfortunately, collecting large-scale high-quality traffic data requires tremendous efforts. Therefore, I collaborated with junior students in our lab to build GeoSparkSim [2], a scalable traffic simulator in GeoSpark which extends Apache Spark to generate large-scale road network traffic data with microscopic traffic models such as traffic lights, lane changing, and car following.

GeoSparkSim converts road networks and simulated vehicles to Spatial RDDs. Then it parallelizes each step in traffic simulation into a set of RDD transformations and distributes the computation-intensive microscopic simulation workload to every machine in a cluster. It also employs a simulation-aware spatial-temporal partitioning method to partition data among different machines such that each machine takes a roughly similar amount of simulation workload to achieve load balance. Results of this work were published in MDM 2019 [2] and won Best Demo Paper Runner-Up Award in SSTD 2019.

1.2 System support for interactive spatial visualization dashboards

When a user explores a spatial dataset using a visualization dashboard, such as Tableau and ArcGIS, it often involves several interactions between the dashboard and underlying databases. In each interaction, the dashboard application first issues a query to extract the data of interest from the underlying data system, and then runs the visual analysis task (e.g., heat maps and statistical analysis) on the selected data. The same user may iteratively go through such steps several times to draw useful insights from a dataset. Every iteration may take a significant amount of time to run when dealing with large-scale data. To remedy that, there are two kinds of approaches used by practitioners: (1) draw a small sample of the entire data table and run the dashboard on this subset (2) run SQL queries over the entire table for every interaction, draw a sample of the extracted population and send it back to the dashboard. Unfortunately, the former cannot provide deterministic accuracy loss while the latter suffers from non-negligible query latency and sampling overhead.

In this project, I designed and built Tabula, a middleware that sits between the data system and the geospatial visualization dashboard to uphold interactive spatial visual analytics (see Figure 1). The proposed system adopts a materialized sampling cube approach, which quickly and selectively pre-materializes sampled answers for a set of potentially unforeseen queries (represented by an OLAP cube cell). In each dashboard interaction, the system returns a materialized sample for the SQL

query, rather than the original query answer. This approach mitigates the data-to-visualization time since the pre-materialized samples significantly reduce both query time and visualization time. Tabula allows data scientists to define their own accuracy loss function that suits specific domains, such as geospatial heatmap analysis or statistical analysis. In the meantime, the system guarantees that accuracy loss never exceeds a user-specified accuracy loss threshold. I implemented a prototype of Tabula in an open-source distributed data system, Apache Spark. A research paper featuring Tabula is directly accepted to ICDE 2020 without revision [5] (3% direct acceptance rate).

1.3 Lightweight data indexing for big data systems

Spatial data management systems such as PostGIS and GeoSpark often employ index structures, e.g., R-Tree (GeoSpark uses R-Tree as a part of its distributed index), to speed up SQL queries on the indexed table. Even though classic database indexes reduce the query response time, they face the following challenges: (1) A database index usually yields 5% to 15% additional storage overhead (e.g., Solid State Drives, Non-Volatile Memory and Hard Disk Drive). Although the overhead may not seem too high in small databases, it results in non-ignorable dollar cost in big data scenarios. (2) Maintaining a database index incurs high latency because the DBMS has to locate and update those index entries affected by the underlying table changes. Many widely-used spatial or non-spatial indices including B+Tree, R-Tree, and Quad-Tree are encumbered by these challenges.

Lightweight database indexing. To tackle that, I designed Hippo index [3, 4], a fast, yet scalable, database indexing approach. It significantly shrinks the index storage and mitigates maintenance overhead without compromising much on the query execution performance. Hippo stores disk page ranges instead of tuple pointers in the indexed table to reduce the storage space occupied by the index. It maintains simplified histograms that represent the data distribution and adopts a page grouping technique that groups contiguous pages into page ranges based on the similarity of their index key attribute distributions. When a query is issued, Hippo leverages the page ranges and histogram-based page summaries to recognize those pages such that their tuples are guaranteed not to satisfy the query predicates and inspects the remaining pages.

A prototype of Hippo index is implemented inside an existing open-source database management system PostgreSQL and the code is released for public use[1]. Two research papers about Hippo, one for regular data and one for spatial data, were published in PVLDB 2016 [3] and SSTD 2017 [4], respectively. In addition, I implemented a spatial analysis application - taxi pickup point heat map - using Hippo as the back-end to demonstrate the performance in ICDE 2017.

ML-enhanced secondary index using correlation. During my internship at IBM Almaden Research Center, I collaborated with IBM researchers and designed Hermit, a succinct secondary indexing mechanism that judiciously leverages the rich soft functional dependencies hidden among columns for indexed key access. Instead of building a complete index (e.g., B^+-Tree) of the key column, Hermit navigates any incoming key access queries to an existing index built on other correlated columns, namely host index. This navigation is attained through a succinct machine-learning enhanced data structure that models correlation using a tiered regression method. Results of this work were published in SIGMOD 2019 [8] and demonstrated in PVLDB 2019.

Updatable learned index. During another internship at Microsoft Research, I worked with the researchers there on an updatable learned index, namely Alex, that works beyond static data. Alex reorganizes storage structures in index tree nodes to support efficient data insertion. As a side effect, the reorganized storage structures also improve search performance. Moreover, Alex equips an adaptive index structure to gain model robustness when the data distribution shifts. A research paper about Alex was published in SIGMOD 2020 [1].

2 Research Philosophy

Based on my experience as a doctoral student and an intern in several companies, I developed my own research philosophy which, I believe, will foster the new generation of data systems. My philosophy is summarized as follows:

- *System-oriented research.* Building data systems that really work benefits both academia and industry. It will in turn help me discover new research challenges in real-world scenarios. For example, at Arizona State University, after I open-sourced GeoSpark, many researchers and practitioners adopted this system and sent me constructive feedback. This valuable information motivated me to refine my research ideas and helped eventually propose Tabula system.
- *Research collaboration.* Seeking the knowledge from and collaborating with experts in different places is the way to solve and recognize challenging problems with wide applicability. During the course of my PhD studies, I have collaborated with experts from several companies, e.g., Microsoft Research, IBM Almaden Research Center and Apple, as well as junior colleagues at school. These collaborations led to several new systems that were published in top venues.
- *Diversity of research areas.* Working in several research areas gives a broader vision of interdisciplinary opportunities and inspires more practical research ideas. My current interdisciplinary research that connects database systems and GIS is actively engaging collaborators across a range of relevant disciplines such as geography and urban planning.

[1]Hippo source code: github.com/DataSystemsLab/hippo-postgresql, demo video: youtube.com/watch?v=wWaOK2-9k9A

3 Future Research

In the future, I will continue developing open-source high-performance data management systems to make sense of "Big Spatial Data". My current work raises a number of challenging research problems in this direction that I plan to address immediately. Examples of my future research topics are given below:

Large-scale spatial streaming data analytics. The unprecedented popularity of GPS-equipped mobile devices and Internet of Things (IoT) sensors has led to continuously generating large-scale location information combined with the status of surrounding environments. Such data has a streaming nature and keeps evolving at a staggering rate over time. Precisely digesting the massive spatial streaming data that swarms into the database systems in a short time window requires a well-designed system architecture. It will be greatly beneficial to spatial data scientists in a variety of real-world scenarios. For instance, to make timely planning strategies, the city of Chicago started installing sensors across its road intersections to monitor the environment, air quality, and traffic. Furthermore, making sense of real-time streaming data from these "never sleeping" GPS-equipped devices may even bolster autonomous city governance (i.e., City Brain) including AI-based climate control and traffic planning. Unfortunately, existing streaming data management systems are not scalable and efficient to handle spatial streaming data. They either require tedious programming tasks or suffer from a significant performance drop. To remedy that, I plan to address the scalability issue of large-scale spatial streaming data analytics by developing a full-fledged distributed spatial streaming analytics system. To be specific, I am going to investigate the applicability of Spark Streaming and Apache Flink to continuous spatial queries such as range query, join query and nearest-neighbor query. I also plan to carry out research in inventing distributed spatial data structures in the streaming environment.

Interactive visual analysis of dynamic geospatial data. Geospatial visual analytics is the science of analytical reasoning assisted by geo-visual map interfaces. In my current work Tabula (see Section 1.2), I have shown that interactive spatial visual analytics can help users easily find interesting insights. Although there is a flurry of research projects tackling this problem from different angles, the existing work mainly focuses on static data rather than dynamic data, with the latter becoming more popular recently. Consider an example user who wants to set up a real-time heat map of millions of GPS-installed vehicles in New York City (see the heat map in Figure 1). As time goes on, this heat map should change every minute or even every second to reflect the actual movement of vehicles from place to place. Moreover, the user may impose filters on numerical or textual attributes of moving vehicles or zoom in to a particular region for more details. The interactive nature of geospatial visual analytics requires an immediate response from visualization systems. I plan to address several challenges in this topic that include the co-optimization between underlying database systems and front-end visualization frameworks and materialized visualization maintenance.

Machine Learning-enhanced spatial data structures. Machine Learning techniques have introduced significant performance improvement to several classic database components such as data indices and query optimizers. For example, during my previous internships at IBM-Almaden and Microsoft Research, we leveraged ML techniques to learn succinct index structures from data distributions. Therefore, the user can see analysis results with lower storage cost yet at a higher speed. In this ML revolution, spatial data has got little attention and is merely treated as a second-class citizen. However, spatial data has several special features that deserve better designs. First, spatial data usually consists of heterogeneous objects including points, polygons, and trajectories. Second, spatial queries are often clustered to certain hot geographical regions. Third, other attributes in a dataset are commonly correlated to the spatial attribute (e.g., spatial auto-correlation: income - education -> location). Thus, I plan to design a set of ML-enhanced spatial data structures such as indices or new physical data layouts to facilitate spatial query processing. The newly invented spatial data structures can also be embedded into my future projects derived from other research topics that I plan to pursue.

References

[1] J. Ding, U. F. Minhas, **Jia Yu**, C. Wang, H. Zhang, Y. Li, J. Do, D. Kossmann, J. Gehrke, D. Lomet, B. Chandramouli, and T. Kraska. ALEX: An Updatable Adaptive Learned Index. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, page to appear, 2020.

[2] Z. Fu, **Jia Yu**, and M. Sarwat. Building Microscopic Road Network Traffic Simulators in Apache Spark. In *Proceedings of the International Conference on Mobile Data Management, MDM*, pages 320–328, 2019.

[3] **Jia Yu** and M. Sarwat. Two Birds, One Stone: A Fast, yet Lightweight, Indexing Scheme for Modern Database Systems. *Proceedings of the VLDB Endowment, PVLDB*, 10(4):385–396, 2016.

[4] **Jia Yu** and M. Sarwat. Indexing the Pickup and Drop-Off Locations of NYC Taxi Trips in PostgreSQL - Lessons from the Road. In *Proceedings of the International Symposium on Advances in Spatial and Temporal Databases, SSTD*, pages 145–162, 2017.

[5] **Jia Yu** and M. Sarwat. Turbocharging Geospatial Visualization Dashboards via a Materialized Sampling Cube Approach. In *Proceedings of the International Conference on Data Engineering, ICDE*, page to appear, 2020.

[6] **Jia Yu**, Z. Zhang, and M. Sarwat. GeoSparkViz: a scalable geospatial data visualization framework in the Apache Spark ecosystem. In *Proceedings of the International Conference on Scientific and Statistical Database Management, SSDBM*, pages 15:1–15:12, 2018.

[7] **Jia Yu**, Z. Zhang, and M. Sarwat. Spatial Data Management in Apache Spark: the GeoSpark Perspective and Beyond. *GeoInformatica*, 23(1):37–78, 2019.

[8] Y. Wu, **Jia Yu**, Y. Tian, R. Barber, and R. Sidle. Designing Succinct Secondary Indexing Mechanism by Exploiting Column Correlations. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 1223–1240, 2019.