

Chapter 6

Modeling and Indexing Spatiotemporal Trajectory Data in Non-Relational Databases

Berkay Aydin

Georgia State University, USA

Vijay Akkineni

Georgia State University, USA

Rafal A Angryk

Georgia State University, USA

ABSTRACT

With the ever-growing nature of spatiotemporal data, it is inevitable to use non-relational and distributed database systems for storing massive spatiotemporal datasets. In this chapter, the important aspects of non-relational (NoSQL) databases for storing large-scale spatiotemporal trajectory data are investigated. Mainly, two data storage schemata are proposed for storing trajectories, which are called traditional and partitioned data models. Additionally spatiotemporal and non-spatiotemporal indexing structures are designed for efficiently retrieving data under different usage scenarios. The results of the experiments exhibit the advantages of utilizing data models and indexing structures for various query types.

INTRODUCTION

In recent years, the rapid advancements in satellite imagery technology, GPS enabled devices, location-based web services, and social networks caused a proliferation of massive spatiotemporal data sets (Nascimento, Pfoser, & Theodoridis, 2003). Many consumer-oriented applications from social networks (Facebook, Twitter, Swarm) to mobile services including routing (Google Maps, Apple Maps), taxi services (Uber) etc. consume and generate spatio-temporal location data (Quercia, Lathia, Calabrese, Lorenzo, & Crowcroft, 2010). Furthermore, there are many massive spatiotemporal data repositories generated

DOI: 10.4018/978-1-4666-9834-5.ch006

by scientific resources that are monitoring the moving objects. These include solar events (Schuh *et al.*, 2013), animal migrations (Buchin, Dodge, & Speckmann, 2014), and meteorological phenomena (J. J. Wang *et al.*, 2014). Most traditional relational database management systems provide efficient storage and retrieval schema for almost all types of data. However, usually, they are optimized for datasets of gigabytes of size and centralized processing. On the other hand, NoSQL databases, also known as non-relational databases, refer to a set of database systems that emphasize schema-free models, and ad hoc data organization. Many people are increasingly using these databases where scalability, high volume and fault tolerance of *big spatiotemporal data* are key deciding factors. NoSQL databases being an umbrella for several types of data stores such as key-value store, column store, document store, graph database and several other storage formats, there is no *one-for-all* spatiotemporal model used in non-relational databases; and appropriateness of a particular solution depends on the problems to be solved.

Relational database management systems such as *PostgreSQL* (with *PostGIS*), *Oracle* (*Spatial and Graph*) are designed to store, index and query data that represents geometric objects with spatial characteristics. However, because of the computationally expensive (both processing and storage-wise) spatial and spatiotemporal joins, the scalability of the relational databases are restricted. Many modern applications, including real time object tracking and spatiotemporal data analyses require massive amounts of data ingestion, storage, and query streaming. These tasks require a demand for horizontal scalability.

For solving these problems in traditional RDBMS settings, vertical scaling (increase in processing power and memory of an individual processing unit) is needed. In non-relational databases, horizontal scaling (increasing the number of computers/nodes in a distributed system) can be used for addressing such problems. For our work, we have used Apache Accumulo (Sawyer, O’Gwynn, Tran, & Yu, 2013), which is one of the popular column-based non-relational databases with notable features such as load balancing, horizontal scalability, automatic table partitioning (which will be presented in detail in *Related Work* section). Accumulo also provides custom server-side iterators that can be efficiently utilized when performing spatiotemporal operations needed in queries involving spatiotemporal predicates.

Specifically, in this work, we have approached the problem of storing massive trajectory-based spatiotemporal data in the context of non-relational databases. One part of the problem is the representation of a spatiotemporal trajectory to fit the underlying storage system. Before all else, the design of data models for storing spatiotemporal trajectories in key-value stores is presented. For comparison purposes, our first class of data models (traditional data model) mimics the traditional object-relational database organization. On the other hand, our second class of data models (partitioned data model) exploits the sorted nature of row identifiers in Accumulo database, and stores data using the identifiers of spatiotemporal partitions. For increasing the query performance of proposed data models for different scenarios, in-memory indexing structures are also designed. Further discussion of the indexing structures can be found in *Types of Queries and Indexing Trajectories* section.

The rest of this chapter is organized as follows. In *Related Work* section, related work on storing and indexing spatiotemporal trajectories in traditional (single machine) and distributed storage systems is presented. In *Non-relational Databases* section, background information on distributed non-relational databases with a focus on Accumulo database will be provided. In *Modeling Trajectories in Non-relational Databases*, our data models for storing trajectories are demonstrated. Next, the different types of queries and the indexing strategies for increasing the query performances are shown. Lastly, we present our experimental evaluation; and point future directions; then, conclude the paper.

28 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/modeling-and-indexing-spatiotemporal-trajectory-data-in-non-relational-databases/145594

Related Content

Predictive Modeling for Imbalanced Big Data in SAS Enterprise Miner and R

Son Nguyen, Alan Olinsky, John Quinnand Phyllis Schumacher (2018). *International Journal of Fog Computing* (pp. 83-108).

www.irma-international.org/article/predictive-modeling-for-imbalanced-big-data-in-sas-enterprise-miner-and-r/210567

Cloud Computing for BioLabs

Abraham Pouliakis, Aris Spathis, Christine Kottaridi, Antonia Mourtzikou, Marilena Stamouli, Stavros Archondakis, Efrossyni Karakitsouand Petros Karakitsos (2014). *Cloud Computing Applications for Quality Health Care Delivery* (pp. 228-249).

www.irma-international.org/chapter/cloud-computing-for-biolabs/110437

Biometric Authentication for Cloud Computing

Tarunpreet Bhatiaand A. K. Verma (2015). *Handbook of Research on Securing Cloud-Based Databases with Biometric Applications* (pp. 209-235).

www.irma-international.org/chapter/biometric-authentication-for-cloud-computing/119345

Classification of File Data Based on Confidentiality in Cloud Computing Using K-NN Classifier

Munwar Ali Zardariand Low Tang Jung (2019). *Cloud Security: Concepts, Methodologies, Tools, and Applications* (pp. 678-697).

www.irma-international.org/chapter/classification-of-file-data-based-on-confidentiality-in-cloud-computing-using-k-nn-classifier/224600

Cloud Cryptography

Renuka Devi Saravanan, Shyamala Loganathanand Saraswathi Shunmuganathan (2024). *Emerging Trends in Cloud Computing Analytics, Scalability, and Service Models* (pp. 84-104).

www.irma-international.org/chapter/cloud-cryptography/337833