Evaluating the Vertical and Horizontal Read and Write Scalability of MobilityDB

Bachelor's Thesis Exposé

Eryk Karol Kściuczyk

1 Context

Spatiotemporal databases are increasing in popularity due to increase of production of such data by numerous IoT devices. As the need for such databases increases, developers try to reach for open source and familiar solutions. MobilityDB, built on top of spatial extension PostGIS, extends PostgreSQL's capabilities regarding spatiotemporal aspects. Previous research showed that MobilityDB extension can be used in conjunction with Citus, a PosgreSQL extension, which provides horizontal scalability options.[1] While query run times of specific queries have been evaluated[2], the vertical and horizontal scalability of the platform haven't been fully explored and mesasured.

2 Problem Statement

MobilityDB's distributed capabilities, enabled by integration with Citus, claim to enhance scalability. However, the platform's ability to efficiently scale read and write operations under real-world workloads has not been rigorously evaluated. Questions remain about latency, throughput, and scalability under vertical and horizontal configurations when handling realistic spatiotemporal datasets.

3 Solution Approach

This thesis aims to empirically evaluate the scalability of MobilityDB in both vertical and horizontal deployment scenarios. I plan to evaluate it by:

1. Deploying MobilityDB on single-node and multi-node setups.
2. Conducting benchmarks with realistic spatiotemporal data to measure metrics like request latency and write throughput.
3. Analyzing how query options affect scalability and performance in different configurations.
4. Comparing scalability trade-offs between vertical and horizontal approaches.

4 Aspired Implementation

In this thesis I will use BerlinMOD benchmark to stimulate realistic spatiotemporal data and evaluate the performance of MobilityDB combined with Citus on different hardware configurations. I will create different hardware configurations using Amazon Elastic Compute Cloud (Amazon EC2). Vertical scaling will be tested by running MobilityDB without Citus on different hardware levels, as well as running MobilityDB with Citus on a single machine. For horizontal scaling I will test running the solution on multiple number of machines.

[1] Bakli, Sakr and Zimanyi, "MobilityDB: A Mobility Database Based on PostgreSQL and PostGIS," ACM Transactions on Database Systems (TODS), Volume 45, Issue 4, Article No. 19, Pages 1-42, December 6, 2020

[2] Bakli, Sakr, and Zimanyi, "Distributed Mobility Data Management in MobilityDB," Proceedings of the 2020 21st IEEE International Conference on Mobile Data Management (MDM), June 2020