

Efficient processing of high-volume spatial data with Spark

ALEXEY EGOROV, DR. JANNIS JAKOBI

Efficient retrieval of geospatial data is crucial but presents scaling challenges. During our transition from PostgreSQL to Apache Spark, we encountered limitations in spatial indexing. While PostgreSQL's indexing supports efficient queries, this is not directly translatable to Spark. The transition required us to create new strategies for managing and querying spatial data effectively. In this talk, we'll share the challenges we faced and the innovative solutions we implemented to address them.

We are the data team at viadukt, a start-up focused on advancing energy-efficient building modernization across Germany. To support this, we set up a comprehensive geospatial database containing building-related data and developed a processing pipeline that integrates both federal and open-source datasets.

As we transitioned from PostgreSQL to Apache Spark, we needed new, efficient approaches for spatial data processing. PostgreSQL's spatial indexing is key to its performance, using bounding boxes around geometries to pre-filter matches before exact calculations. Apache Sedona, which supports large-scale spatial processing, includes spatial indexing but encounters limitations with larger datasets. Without traditional indexing in Spark, we applied partitioning techniques based on H3 cells or Geohashes.

In this talk, we'll explore how to apply these partitioning methods effectively and why they offer advantages. Additionally, we'll dive into the specific benefits and challenges of this approach for high-volume spatial data and provide examples.