QoS-Aware Big Geospatial Data Processing

Antonio CORRADI, Professor - Paolo BELLAVISTA, Associate Professor - Rebecca MONTANARI, Associate Professor - Luca FOSCHINI, Assistant Professor - ALJAWARNEH Isam Mashhour, PhD Student & Research Fellow - Mobile Middleware Research Group - DISI, University of Bologna, ITALY

**Introduction & Motivation**

**SCENARIO**

In smart cities, citizens are active collectors for data while moving around, generating big geospatial data, useful for monitoring smart city phenomena.

**PROBLEMS**

- No integrated support for geospatial data processing in current big data ecosystems (Spark, Hadoop, etc.)
- No domain-specific Quality of Service (QoS)-aware support for geospatial data processing

**MOTIVATION**

Need for domain-specific QoS-aware integrated solutions for big geospatial data processing

**QoS Requirements:**

- balancing workloads between processing elements;
- preserving data geospatial-locality: objects that are colocated in reality are loaded to same processing elements;
- processing boundary spatial objects: those that are colocated objects (point, polygon, etc.) that reside on borders between partitions

**Objectives**

- Optimizing query performance in various query types:
  - proximity queries
  - complex join queries
- Designing a novel QoS-aware big data partitioning support
- Maximizing system’s performance gain, by trading-off the three requirements that affect big geospatial data processing’s QoS: load balancing, geospatial-locality and boundary spatial objects

**Querying Support for Big Geospatial Data Processing**

We have designed a query-router, which takes a prefix (extracted from key’s geohash) as a query predicate to select appropriate partitions, and to route the request accordingly to only those partitions that contain relevant result-set

**Experimental Results**

**Testing Datasets**: we have used big geospatial data (around 15 million records) collected through ParticipAct1, a project of the University of Bologna (UNIBO) that aims to study the potential cooperation between citizens, leveraging smartphones as a tool for interaction and interconnection.

1 http://participact.unibo.it/

**Proximity Query Performance**

our MapReduce-based implementation outperforms MongoDB’s support (geoNear)

**Fig1: Query Routing**

**Data Partitioning Support for Big Geospatial Data Processing**

Our geospatial data partitioning support provides two different methods:

- **Self-Adaptable Partitioner (SAP)**: calculates new cutting factors for a subsequent running session learning from previous runs. Imagining Earth flattened out, cutting factors are analogous to vertical partitioning line in planar geometry.
  - **Benefits**: balanced load, minimized boundary spatial objects
- **Geospatial-Aware Partitioner (GAP)**: all spatial objects that have the same geohash code’s prefix are routed to the same partition.

- **Benefits**: balanced load, preserved geospatial-locality

**Fig2: Self-Adaptable Partitioner (SAP)**

**Fig3: Geospatial-Aware Partitioner (GAP)**

**Conclusions & Ongoing Works**

**Conclusions**:

- Integrated and domain-dependent partitioning and query optimization are crucial for improving big spatial data processing’s QoS
- Our support trades-off the QoS requirements

**Ongoing Works: SYNTHESIS**

- Improving our query-router: integrating additional methods for an improved routing
- Designing a query-router: including query reformulation and minimizing query complexity (replacing joins with semi joins, etc.)
- Designing additional big geospatial data partitioning methods

**Contacts & Publications**

**Contacts**:

- Isam Mashhour Aljawarneh, Paolo Bellavista, Antonio Corradi, Rebecca Montanari, Luca Foschini

- List of relevant publications: