Simba: Efficient In-Memory Spatial Analytics

Big Spatial Data Analysis at Ease

SELECT poi.id, count(*) as c
FROM poi
DISTANCE JOIN data
ON POINT(data.lat, data.long)
IN CIRCLE_RANGE(Point(poi.lat, poi.long), 3.0)
WHERE POINT(data.lat, data.long)
IN RANGE(Point(24.39, 66.88), Point(49.38, 124.84))
ORDER BY poi.id

Database Query
Table Indexing

Query Processing Workflow

Table Indexing

Partition

Local Index

Global Index

CREATE INDEX idx_name ON R(…, x_m) USE idx_type

DROP INDEX idx_name ON table_name

$k$NN Join -- RKJSpark

• R-tree $k$NN join (RKJSpark)
• For each partition $R_i$, find $S_i \subseteq S, \text{ s.t. } \forall R_i \text{ knn}(r, S_i) = \text{knn}(r, S)$
• Define $c_i$ as the centroid of partition $R_i$
• Take a uniform random sample $S' \subseteq S$, and let $\text{knns}(c_i, S') = \{s_1, \ldots, s_k\}$
• For each partition $R_i$:
  $u_i = \max(r, c_i)$
  $\gamma_i = 2u_i + ||c_i - \sum s_j||$
  $S_i = \{s \in S | \gamma_i \leq \gamma_i\}$

Comparison with Existing Systems

Experimental Results

It is now open sourced at: https://github.com/InitialDLab/Simba