

Simba: Spatial In-Memory Big data Analysis

Dong Xie¹, Feifei Li¹, Bin Yao², Gefei Li², Zhongpu Chen², Liang Zhou², Minyi Guo²

¹University of Utah, ²Shanghai Jiao Tong University



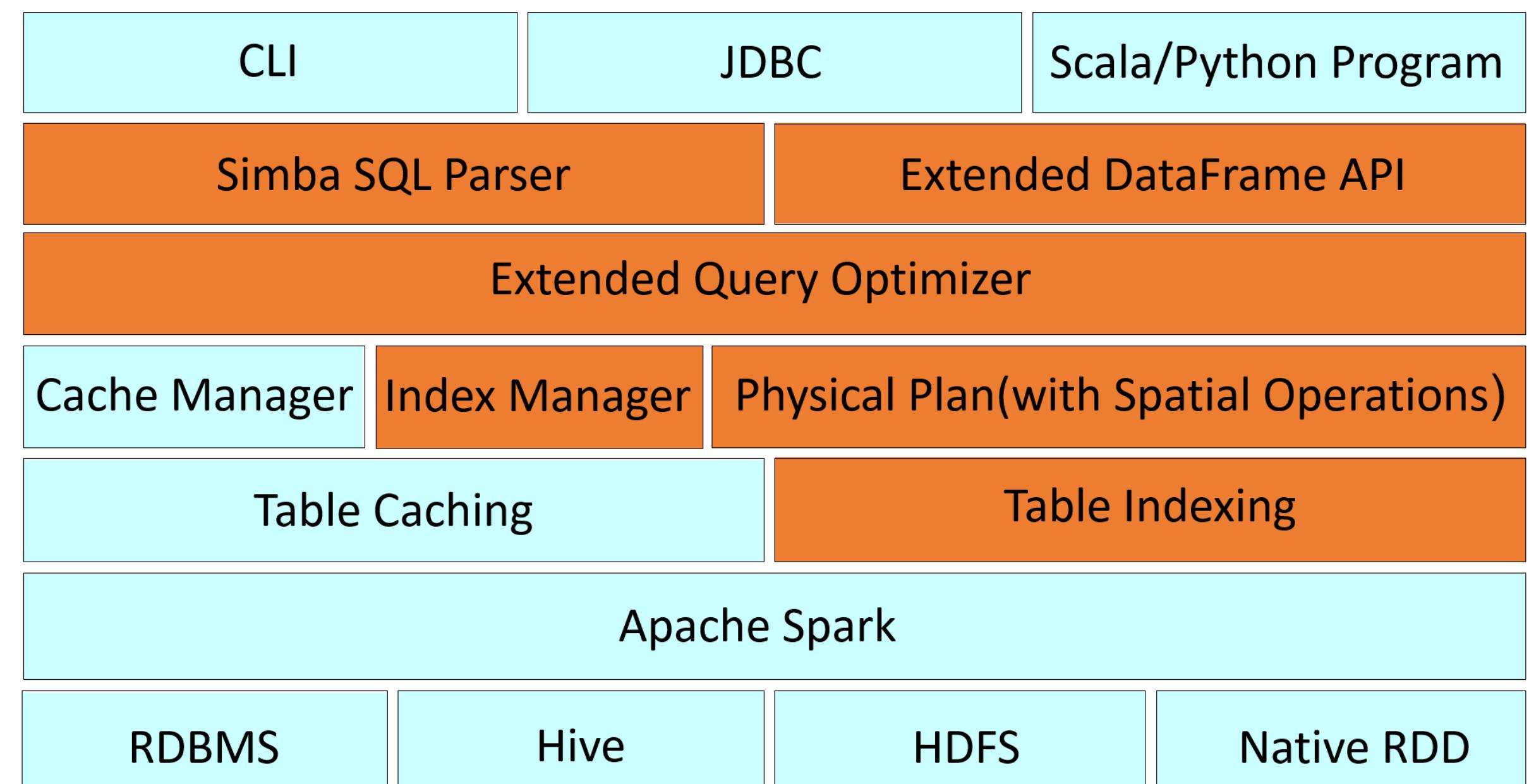
{dongx, lifeifei}@cs.utah.edu {yaoxin@cs., oizz01@, chenzhongpu@, nichozl@, guo-my@cs.}sjtu.edu.cn

Big Spatial Data Analysis at Ease

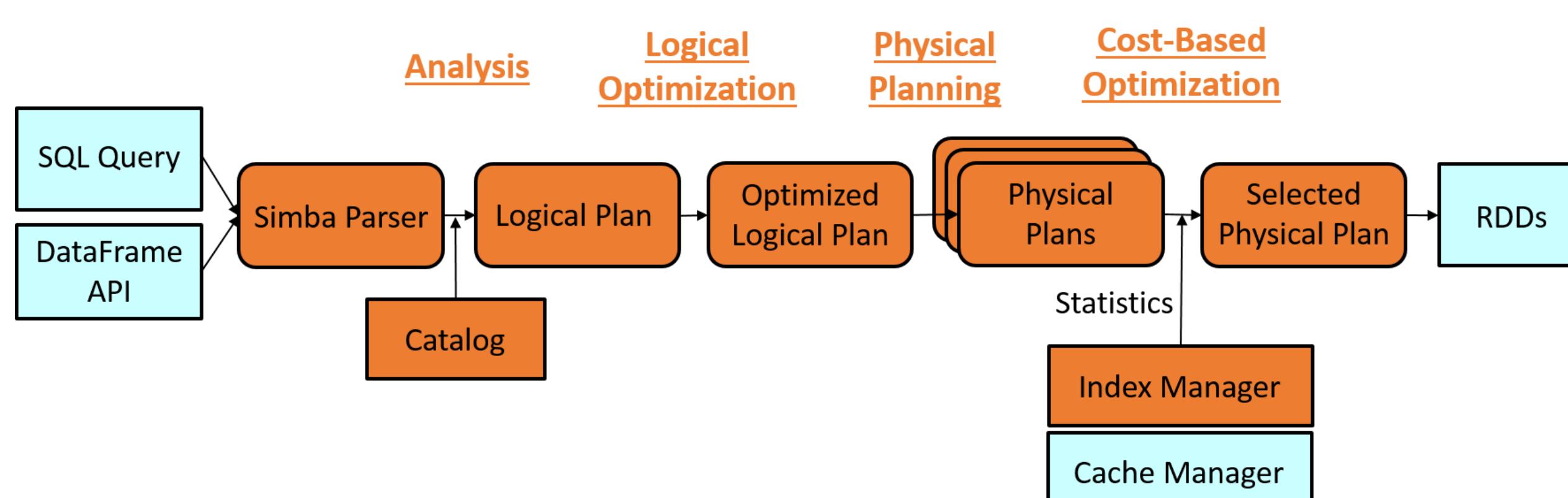
```
SELECT poi.id, count(*) as c
FROM poi DISTANCE JOIN data
  ON POINT(data.lat, data.long)
    IN CIRCLE RANGE(POINT(poi.lat, poi.long), 3.0)
WHERE POINT(data.lat, data.long)
  IN RANGE(POINT(24.39, 66.88), POINT(49.38, 124.84))
GROUP BY poi.id
ORDER BY poi.id
```

```
poi.distanceJoin(data, Point(poi("lat"), poi("long")),
  Point(data("lat"), data("long")), 3.0)
  .range(Point(data("lat"), data("long")),
  Point(24.39, 66.88), Point(49.38, 124.84))
  .groupBy(poi("id"))
  .agg(count("*").as("c")).sort(poi("id")).show()
```

System Architecture



Query Processing Workflow



Range & kNN Query

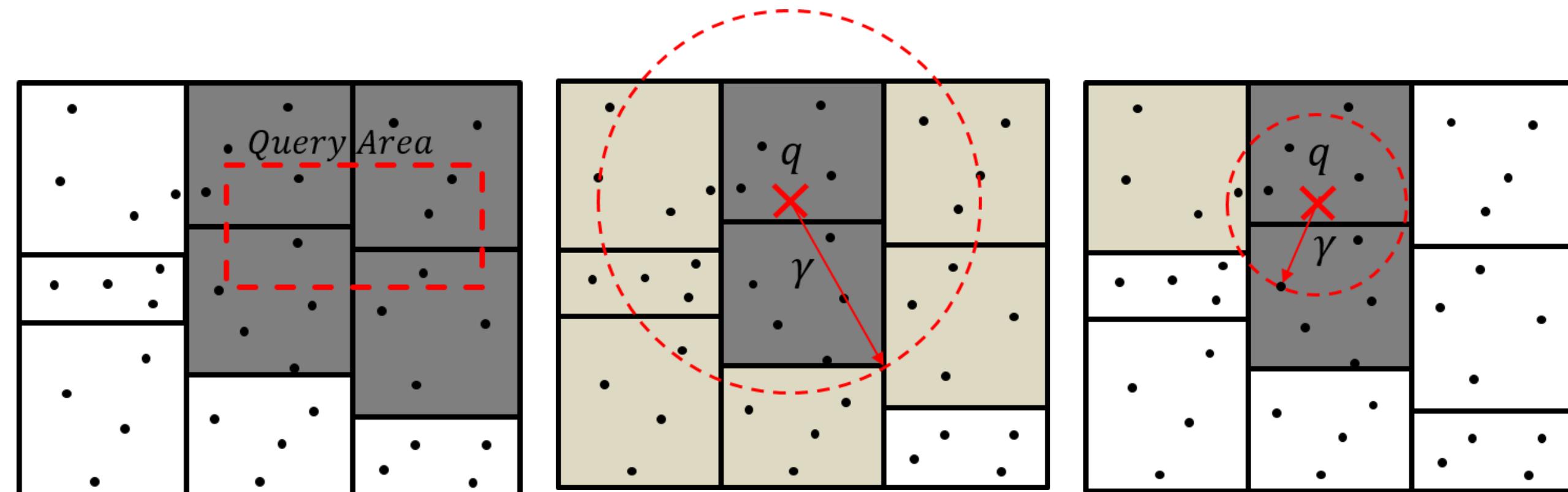
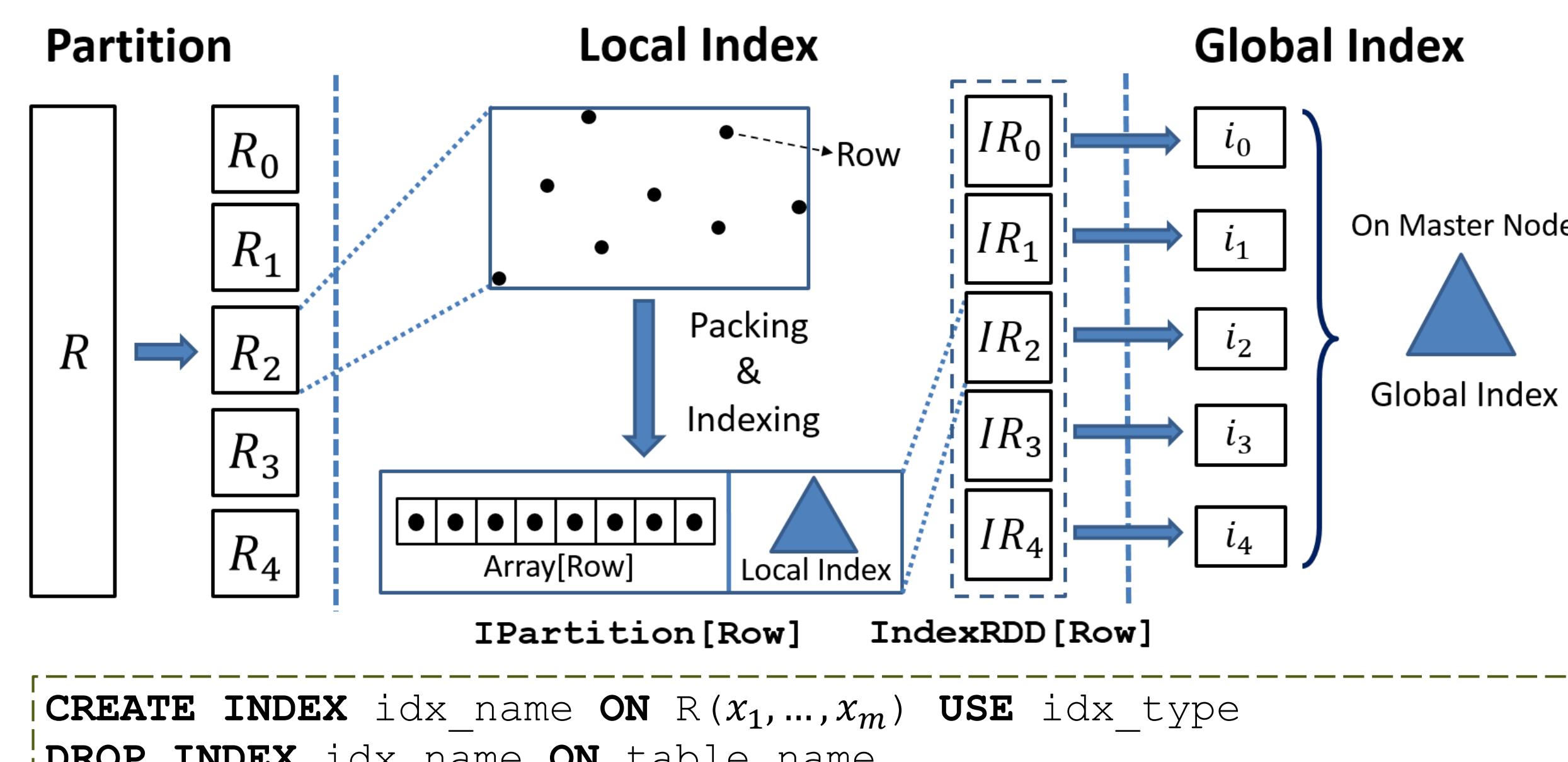
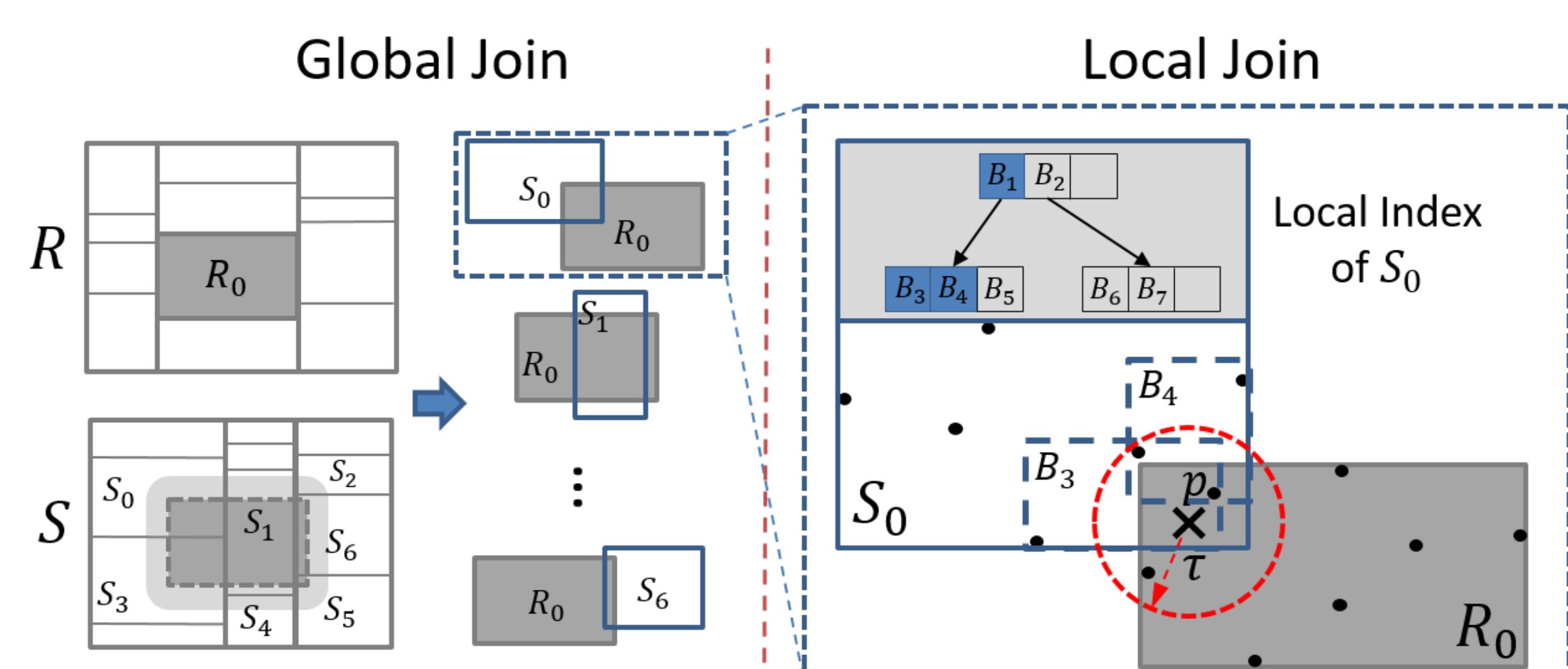


Table Indexing



Distance Join -- DJSpark



kNN Join -- RKJSpark

- R-Tree kNN join (RKJSpark)
- For each partition R_i , find $S_i \subset S, s.t. \forall r \in R_i, knn(r, S) = knn(r, S_i)$
- Define cr_i as the centroid of partition R_i
- Take a uniform random sample $S' \subset S$, and let $knn(cr_i, S') = \{s_1, \dots, s_k\}$
- For each partition R_i :

$$u_i = \max_{r \in R_i} |r, cr_i|$$

$$\gamma_i = 2u_i + |cr_i, s_k|$$

$$S_i = \{s | s \in S, |cr_i, s| \leq \gamma_i\}$$

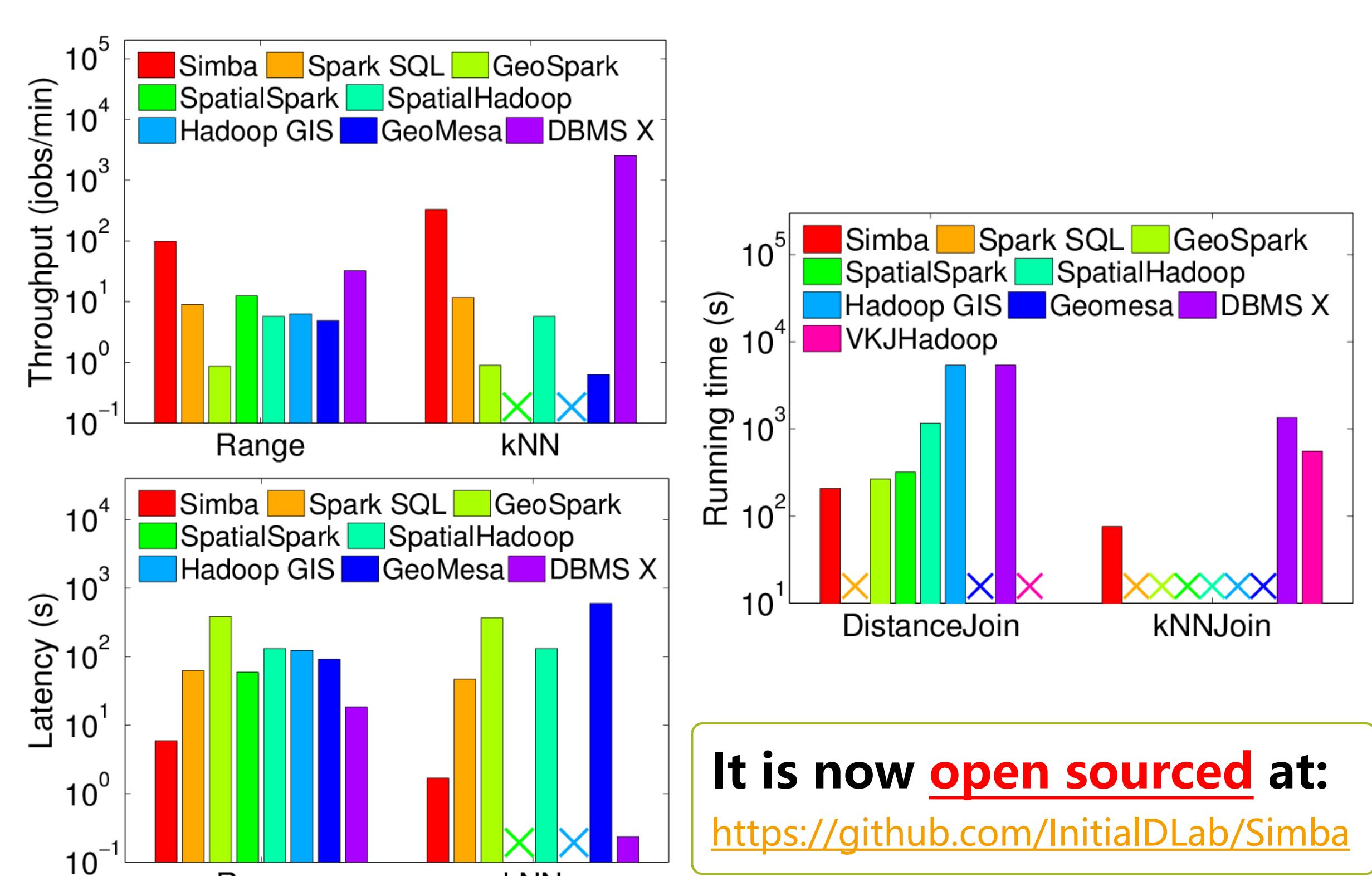
Demo Interface

The demo interface includes a map of Amsterdam with a query point and several red dots representing results. To the right, there is a 'Query' input field containing SQL code, a 'Run Query' button, and a 'Query Result' table showing the retrieved data.

Comparison with Existing Systems

Core Features	Simba	GeoSpark	SpatialSpark	SpatialHadoop	Hadoop GIS
Data dimensions	multiple	$d \leq 2$	$d \leq 2$	$d \leq 2$	$d \leq 2$
SQL	✓	✗	✗	Pigeon	✗
DataFrame API	✓	✗	✗	✗	✗
Spatial indexing	R-tree	R-/quad-tree	grid/kd-tree	grid/R-tree	SATO
In-memory	✓	✓	✓	✗	✗
Query planner	✓	✗	✗	✓	✗
Query optimizer	✓	✗	✗	✗	✗
Concurrent query execution	thread pool in query engine	user-level process	user-level process	user-level process	user-level process
query operation support					
Box range query	✓	✓	✓	✓	✓
Circle range query	✓	✓	✓	✗	✗
k nearest neighbor	✓	✓	only 1NN	✓	✗
Distance join	✓	✓	✓	via spatial join	✓
k NN join	✓	✗	✗	✗	✗
Geometric object	In progress	✓	✓	✓	✓
Compound query	✓	✗	✗	✓	✗

Experimental Results



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