Simba: Efficient In-Memory Spatial Analytics

Dong Xie[1], Feifei Li[1], Bin Yao[2], Gefei Li[2], Liang Zhou[2], Minyi Guo[2]
[1]University of Utah, [2]Shanghai Jiao Tong University
{dongx, lifeifei}@cs.utah.edu {yaobin@cs., oizz01@, nichozl@, guo-my@cs.}sjtu.edu.cn

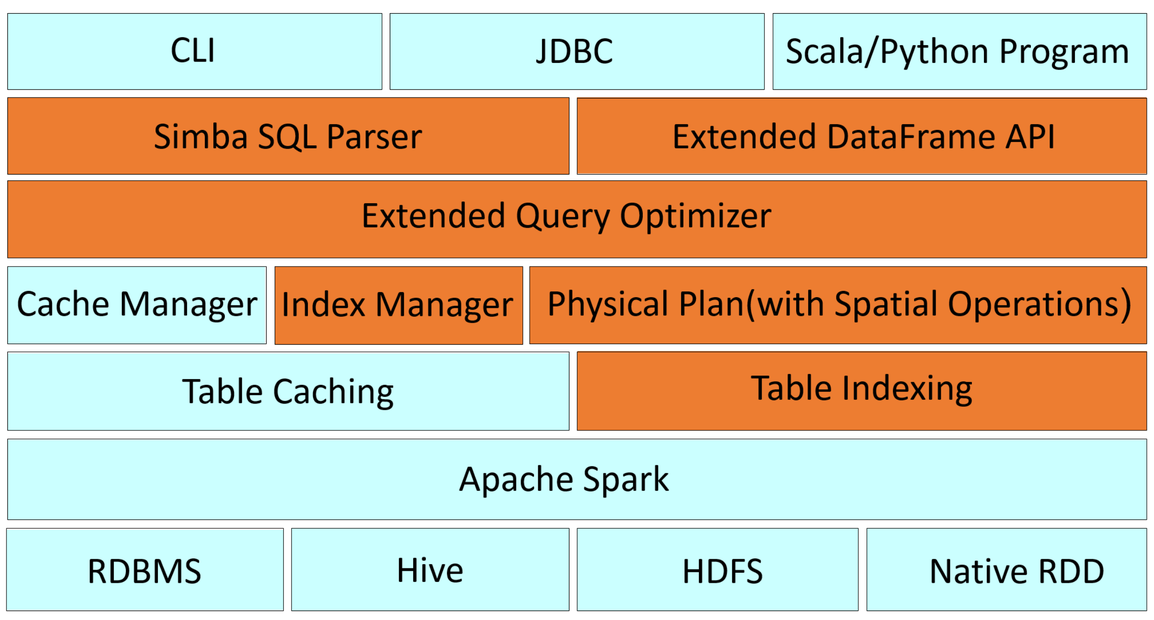
Simba: Spatial In-Memory Big data Analytics

Big Spatial Data Analysis at Ease

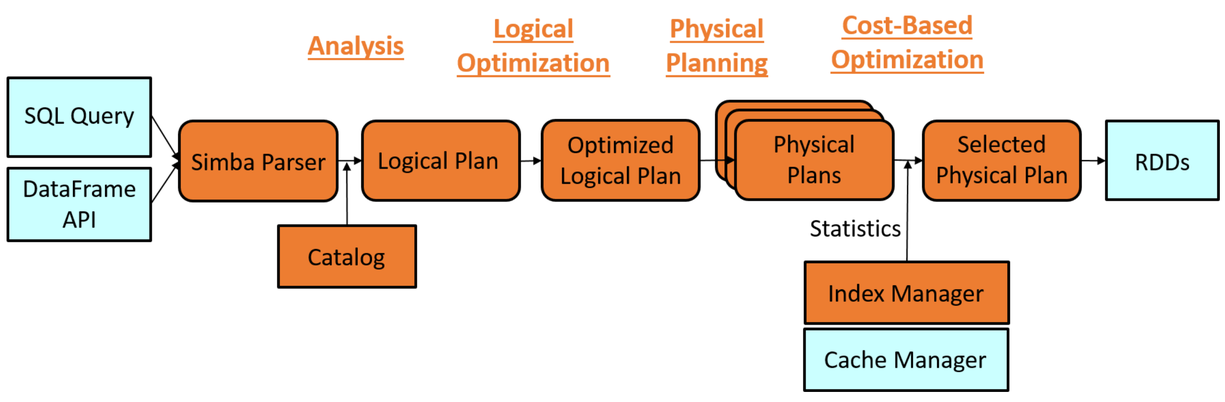
```
SELECT poi.id, count(*) as c
FROM poi DISTANCE JOIN data
     ON POINT(data.lat, data.long)
        IN CIRCLERANGE(POINT(poi.lat, poi.long), 3.0)
WHERE POINT(data.lat, data.long)
      IN RANGE(POINT(24.39, 66.88), POINT(49.38, 124.84))
GROUP BY poi.id
ORDER BY poi.id
```

```
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

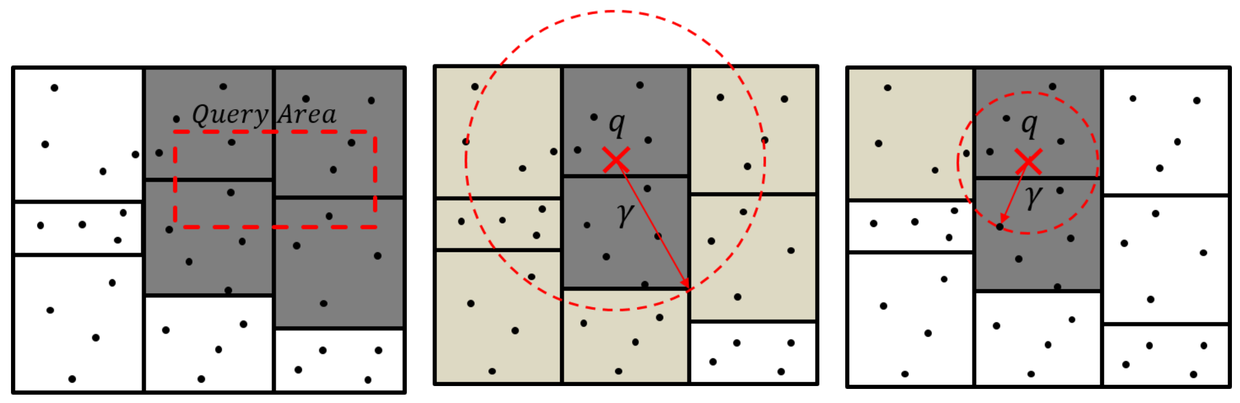
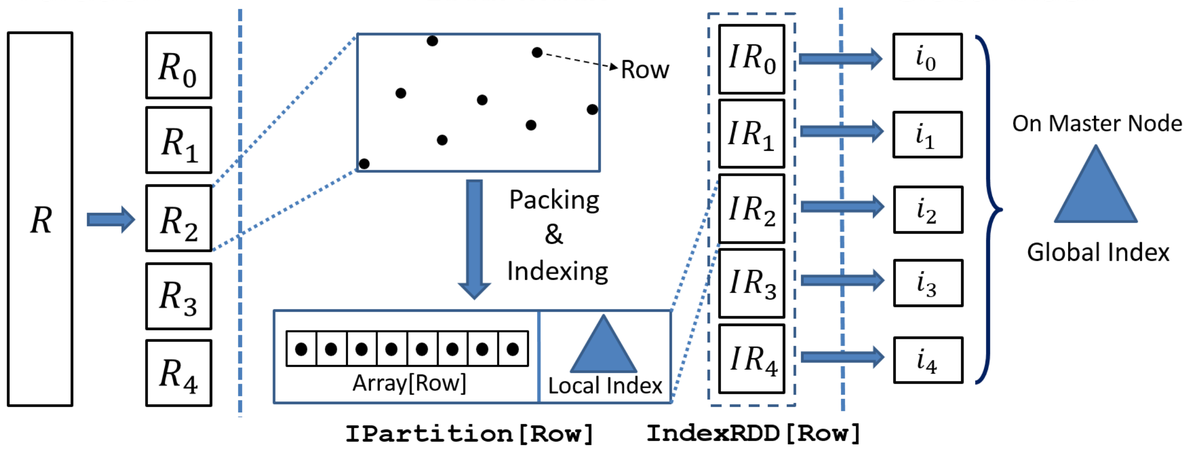


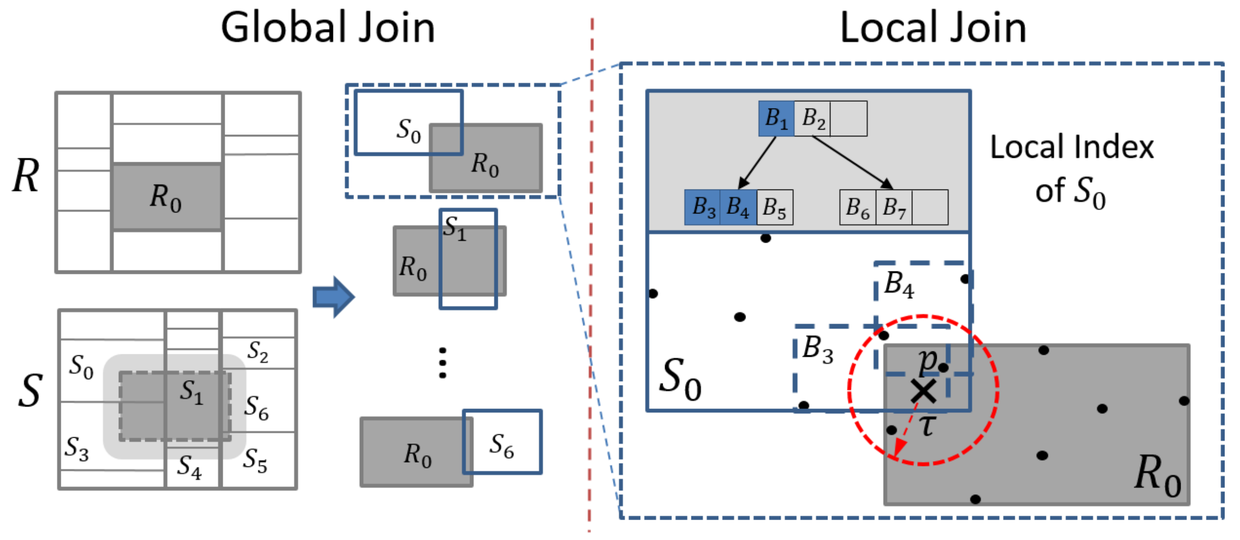
Table Indexing

Partition **Local Index** **Global Index**




```
CREATE INDEX idx_name ON R(x_1, ..., x_m) USE idx_type
DROP INDEX idx_name ON table_name
```

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 $knn(cr_i, S') = \{s_1, \ldots, s_k\}$
- For each partition R_i :

$$u_i = \max_{r \in R_i} |r, cr_i|$$
$$\gamma_i = 2u_i + |cr_i, s_k|$$
$$S_i = \{s | s \in S, |cr_i, s| \leq \gamma_i\}$$

Query Optimizations

- Partition size auto-tuning : predefined **customized partitioner**
- Spatial **predicates merging**

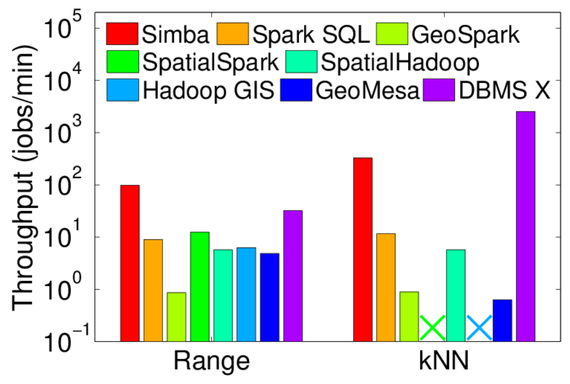
- Index scan optimization : **table scan** → **index look up**
- **Selectivity estimation cost-based Optimization: scan or index**

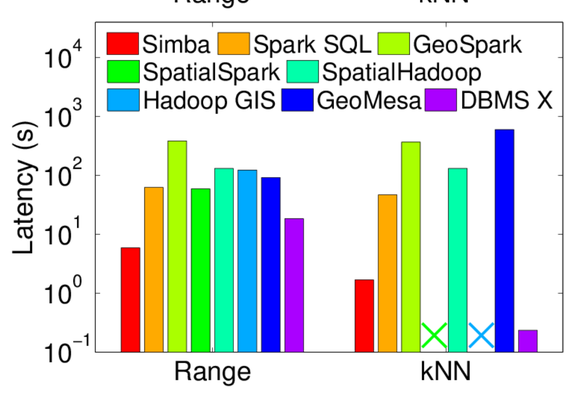
- Broadcast join optimization: **small table joins large table**
- Logical partitioning optimization for **RKJSpark**
  - Provides **tighter pruning bound** γ_i

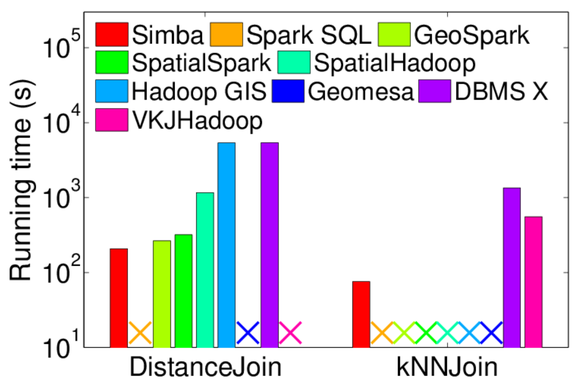
Comparison with Existing Systems

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	✓
kNN join	✓	×	×	×	×
Geometric object	In progress	✓	✓	✓	✓
Compound query	✓	×	×	✓	×

Experimental Results







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