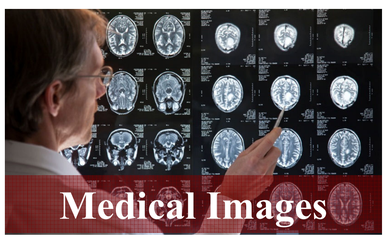
SpatialHadoop
A MapReduce Framework for Spatial Data

http://spatialhadoop.cs.umn.edu/

Ahmed Eldawy Mohamed F. Mokbel
University of Minnesota
shadoop@cs.umn.edu

Big Spatial Data



- Hundreds of millions of smart phones sold every year
- Peta bytes of space data and medical images collected yearly

There's a need for an efficient framework for large scale spatial data handling

Elegant and Powerful

Range query in Hadoop

```
points = LOAD 'points' AS (id:int, x:int, y:int);
result = FILTER points BY
  x < x2 AND x > x1 AND
  y < y2 AND y > y1;
```

Takes 193 seconds

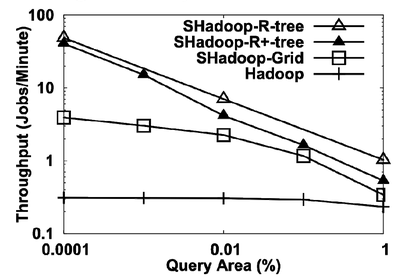
Range query in SpatialHadoop

```
points = LOAD 'points' AS (id:int, location:point);
result = FILTER points BY
  Overlap(location,
      rectangle(x1, y1, x2, y2));
```

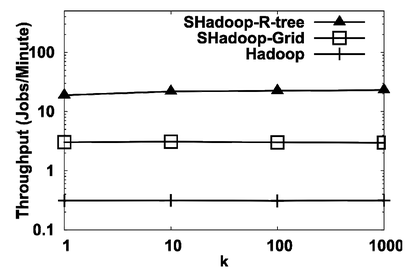
Finishes in 2 sec

Results

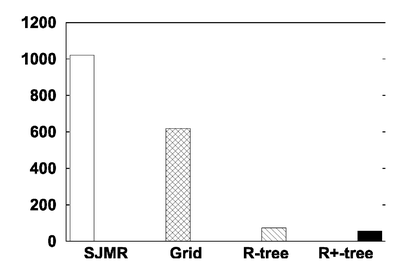
Range Query



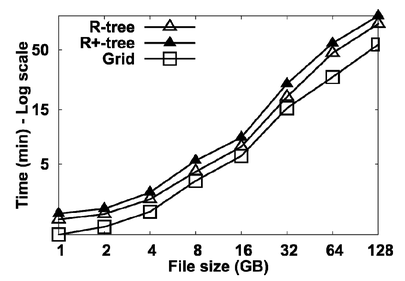
kNN



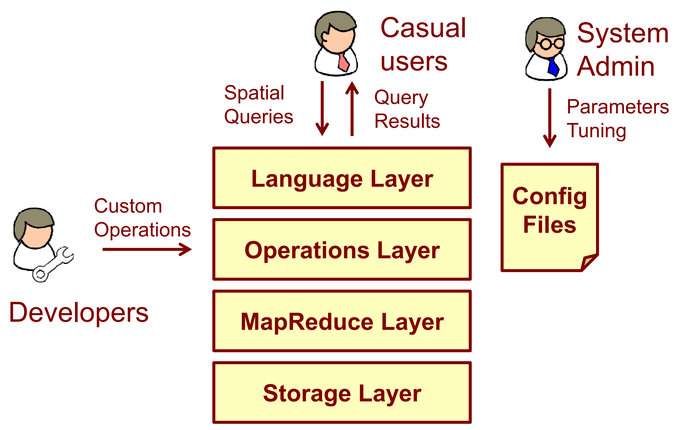
Spatial Join



Index Building



SpatialHadoop Architecture



Language Layer

- OGC standard spatial data types

```
lakes = LOAD 'lakes'
    AS (id:int, boundary:Polygon);
```

```
Point
Linestring
MultiLinestring
Polygon
MultiPolygon
GeometryCollection
```

- Primitive functions

```
Lakes_with_area = FOREACH lakes
  GENERATE id, Area(boundary);
```

```
MakePoint, MakeRectangle
Intersection, Difference
Area, Length, Distance
Overlap, Touches
Union, ConvexHull
```

- Spatial operations

```
lakes_states = JOIN lakes BY lakes_boundary
  states BY states_boundary PREDICATE = OVERLAP
```

```
nearest_houses = KNN houses WITH_K=100
  USING Distance(house_loc, query_loc);
```

How to Install

Tap your NFC-enabled phone here

OR

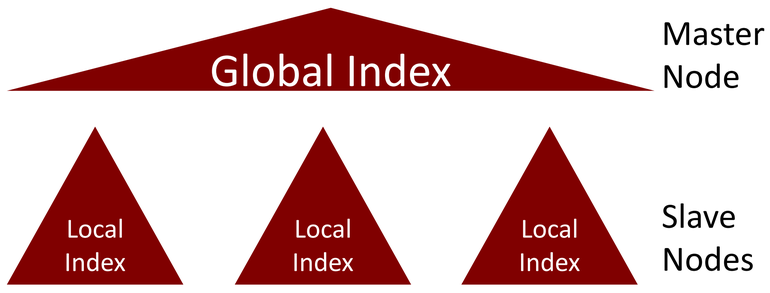
Scan the QR code below

Install in three easy steps

❶Download ❷Configure ❸Run

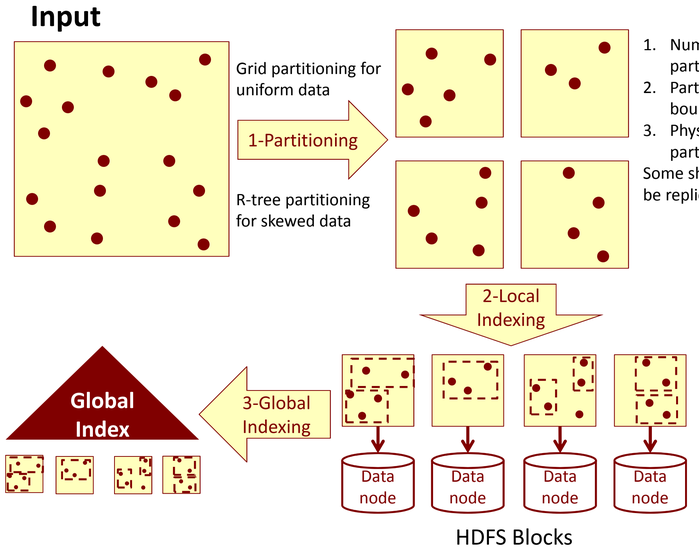
Storage Layer (Indexing)

Hadoop is designed to work with heap non-indexed files
Traditional spatial indexes are designed for:
- Procedural programming while Hadoop uses MapReduce programming
- Traditional file system while Hadoop uses Hadoop Distributed File System (HDFS)
- A two layered approach (Global and local indexes) is used to build grid file, R-tree and R+-tree indexes in SpatialHadoop

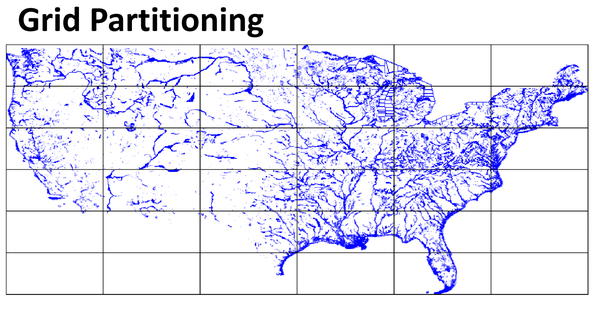


Index Building

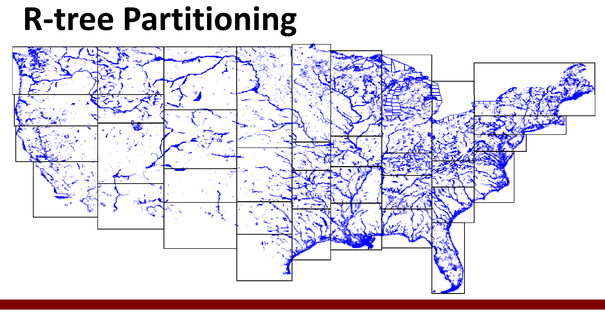
Input



Grid Partitioning



R-tree Partitioning

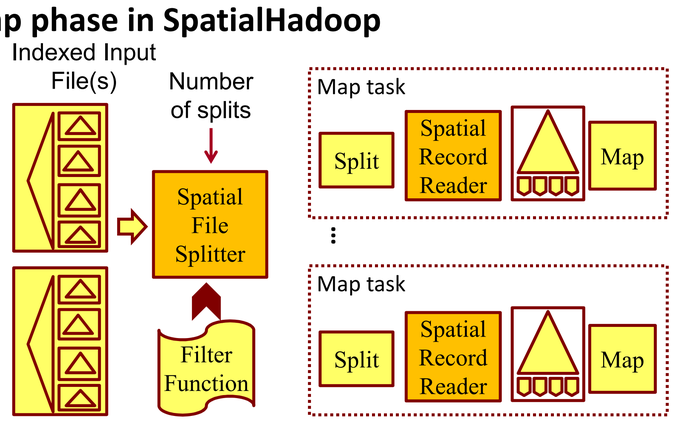


MapReduce Layer

MapReduce programs in Hadoop can only deal with non-indexed files
SpatialHadoop adds two new components to allow MapReduce programs to utilize spatial indexes

1. **SpatialFileSplitter: Utilizes the global index by pruning file blocks that do not contribute to answer**
2. **SpatialRecordReader: Utilizes local indexes by efficiently selecting records that need to be processed**

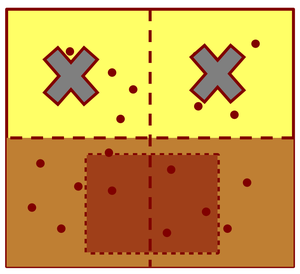
Map phase in SpatialHadoop



Operations Layer

Range query

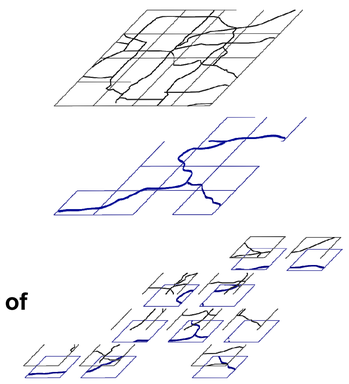
SpatialFileSplitter prunes blocks outside query range
SpatialRecordReader passes local indexes to the map function
Map function selects records in range



Spatial Join

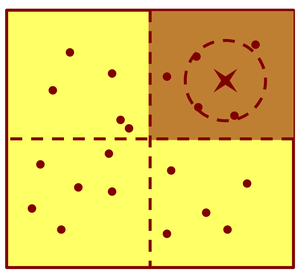
SpatialFileSplitter finds overlapping block pairs

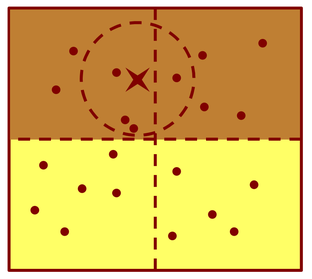
Map function joins each pair of overlapping blocks



kNN

SpatialFileSplitter selects the block that contains the query point
Map function performs kNN in the selected block
Answer is tested for correctness
✓ Initial answer is correct





First iteration runs as before and result is tested for correctness
× Initial answer is incorrect
Second iteration processes other blocks that might contain an answer
✓ Final answer is correct

University of Minnesota
This work is supported in part by the National Science Foundation under Grants IIS-0952977 and IIS-1218168