Optimizing Systems for Geolocation Data: The Works!

By: Mohamed Sarwat

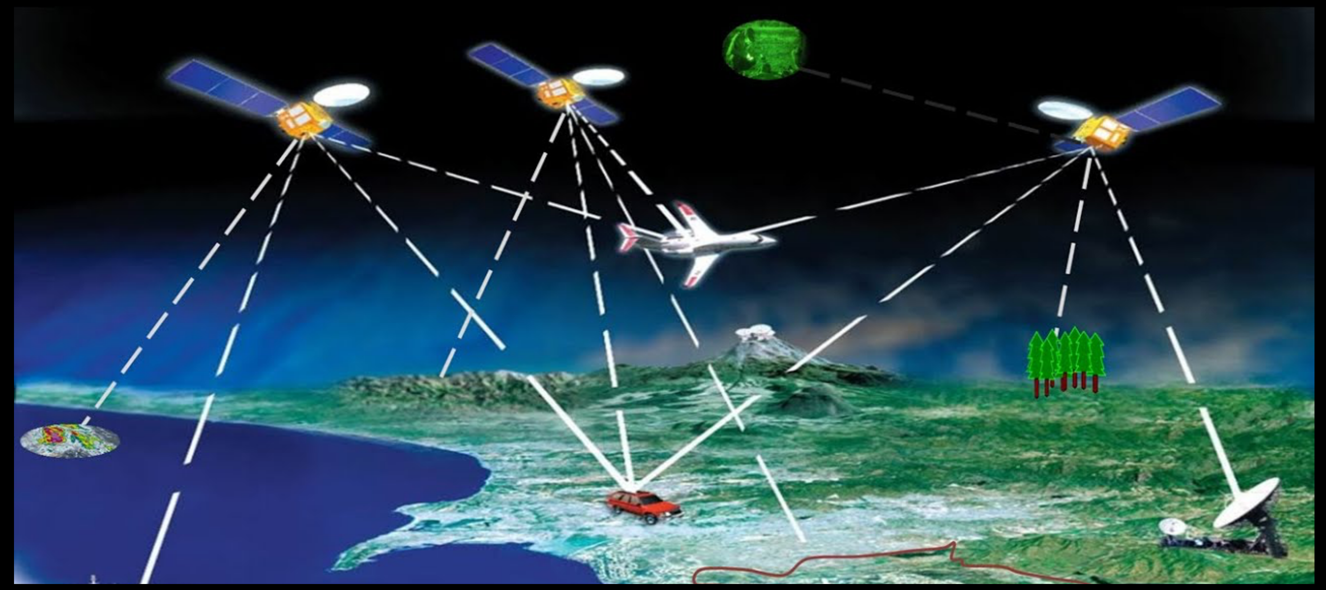
@MoSarwat

Arizona State University

Turning points for our community

- The idea of having a computing/ communication device that is turned on all the time and that you can move around with

- Localization technology (e.g., GPS signal, WiFi signal, Buetooth Signal…)



- Huge Announcement on July 10th 2008

Turning Points for our Community

- Almost all apps provide some sort of location-based services

Apps Generate tons of Geospatial Data

~ 15 million **rides per day**

~ 10 million **rides per day**

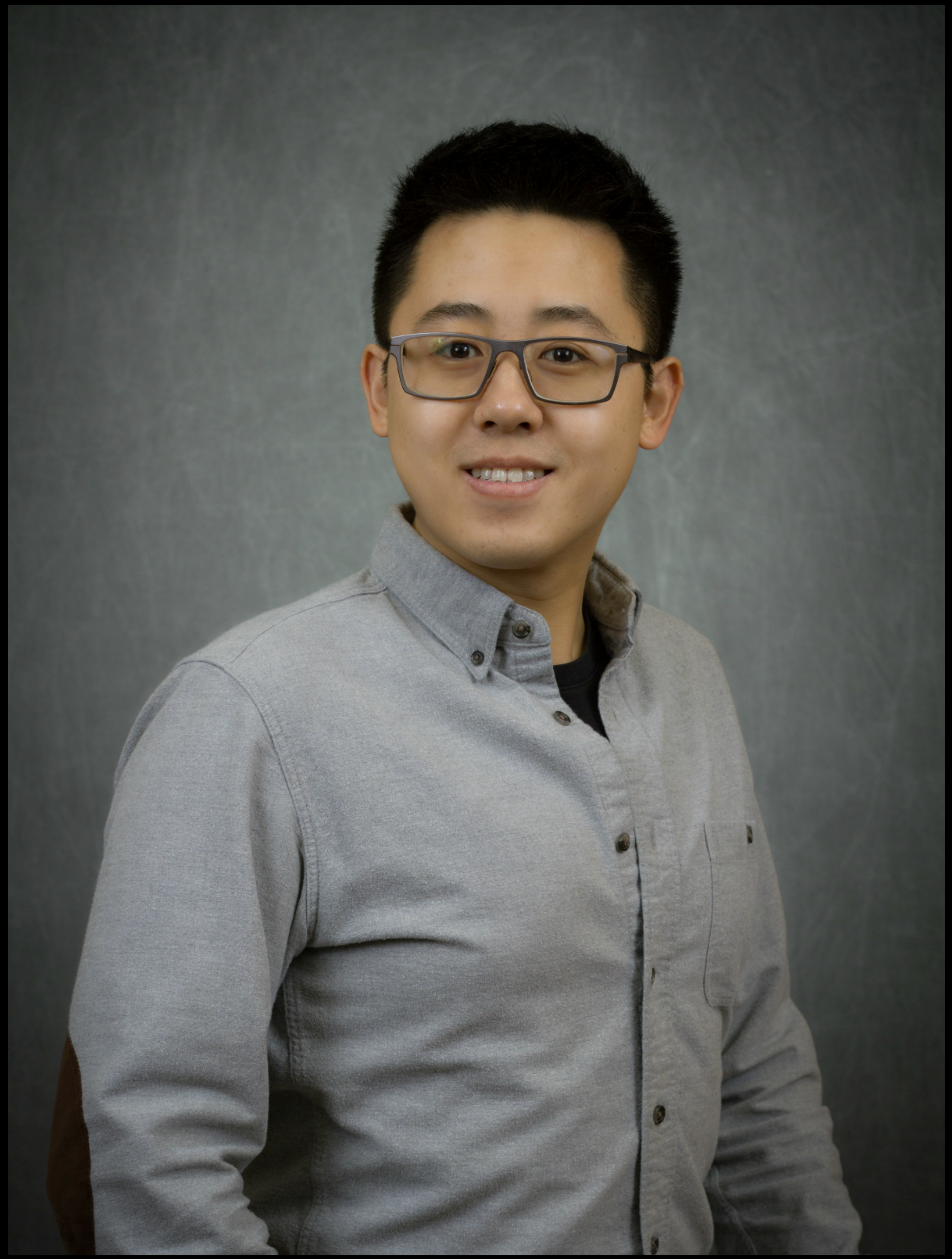
How can we run geospatial computation efficiently on such massive data?

Spark back then…

- **Did not provide a geospatial data processing API**
  - Geospatial was treated as yet another attribute
  - The Geospatial data was only perceived as two extra columns (X,Y)
  - A programmer had to write 1000s of lines of code to implement simple geospatial operations like Spatial Join
- **Could not efficiently optimize geospatial queries**
  - No spatial proximity-aware load balancing
  - No spatial indexes
  - No spatial data compression techniques

Call for Action

- Jia joined the Data Systems Lab in 2015
- I explained the problem to him
- We sat down and came up with the concept of Spatial RDD
- He disappeared for a few weeks and BAM he came back with an initial prototype
- Graduating ;) ;)



Scala API

Spatial SQL

GeoViz API

GeoSimulation API

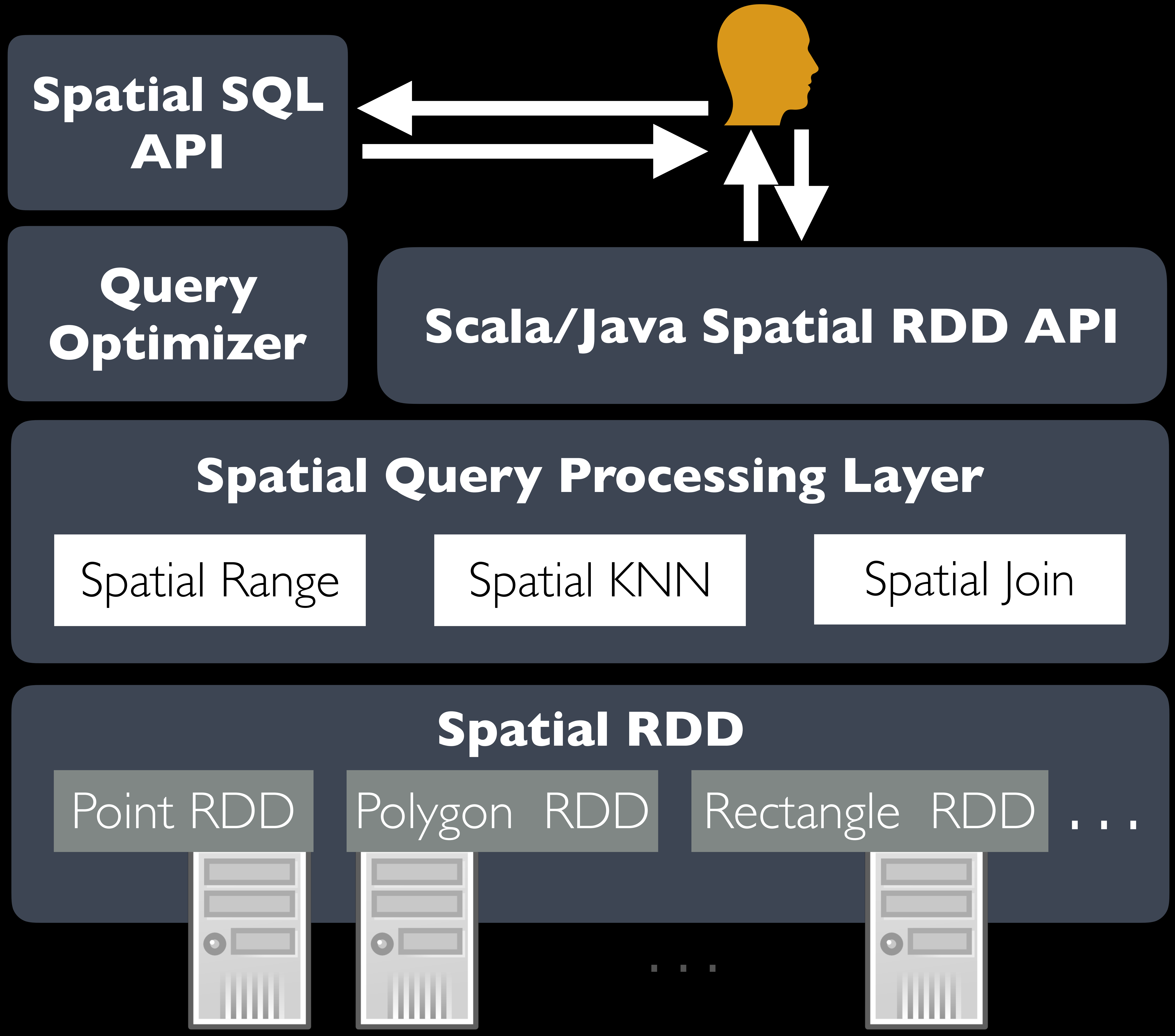
GeoSpark

ESRI ShapeFiles

GeoJSON Docs

NASA HDF/ NetCDF

PostGIS DB



Jia Yu, Jinxuan Wu, ,M. Sarwat, "GeoSpark: a cluster computing framework for processing large-scale spatial data" in ACM SIGSPATIAL GIS 2015

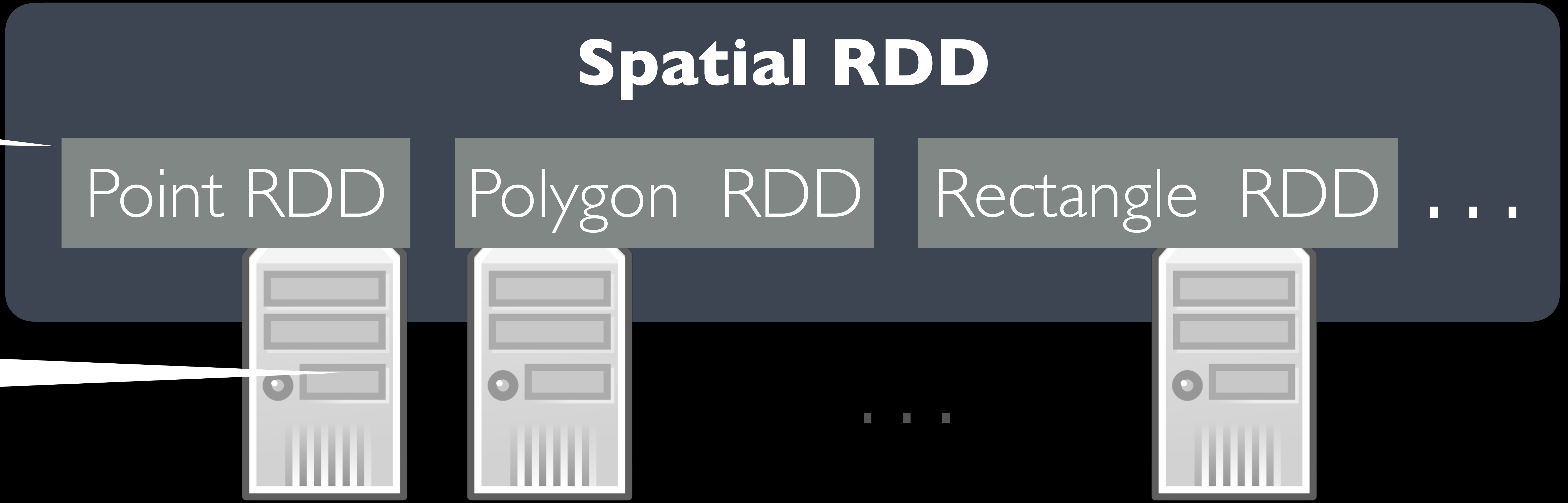
Compact In-Memory Representation

For a point object, GeoSpark takes 20 bytes, Kryo serializer in Spark takes 401 bytes, Java serializer in Spark takes 1170 bytes. On average, GeoSpark custom serializer uses **5-6 times fewer bytes** then default Kryo serializer and **10+ times fewer bytes** then default Java serializer.

1	2	3	4	5-12	13-20	21-28	29-36	...	...	...	...
Object type	Sub-obj num	Sub-obj1 type	Coord num	longitude	latitude	longitude	latitude	...	Sub-obj2 type	Coord num	...

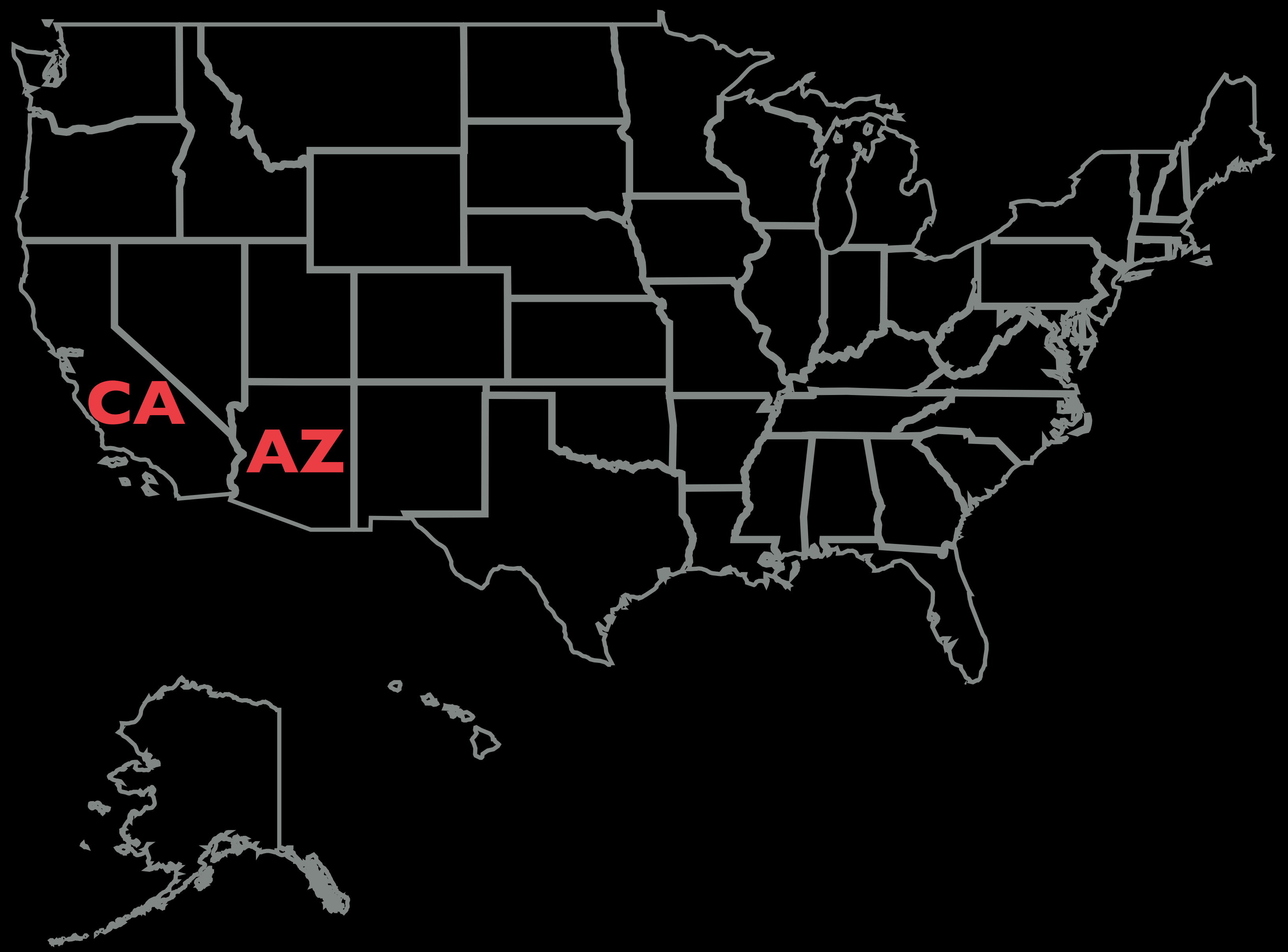
Support heterogenous Spatial data Types

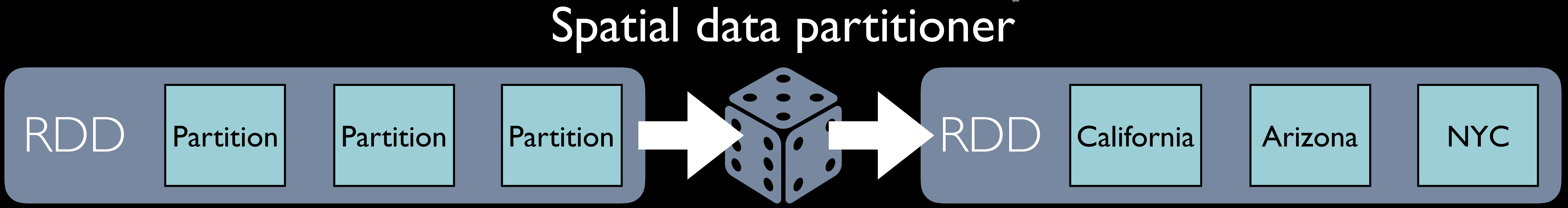
Data is cached in the main memory of the cluster



Spatial Data Partitioning

- Repartition data in RDD
  - Partition by spatial proximity
  - Still achieve load balance
  - API: CustomPartitioner

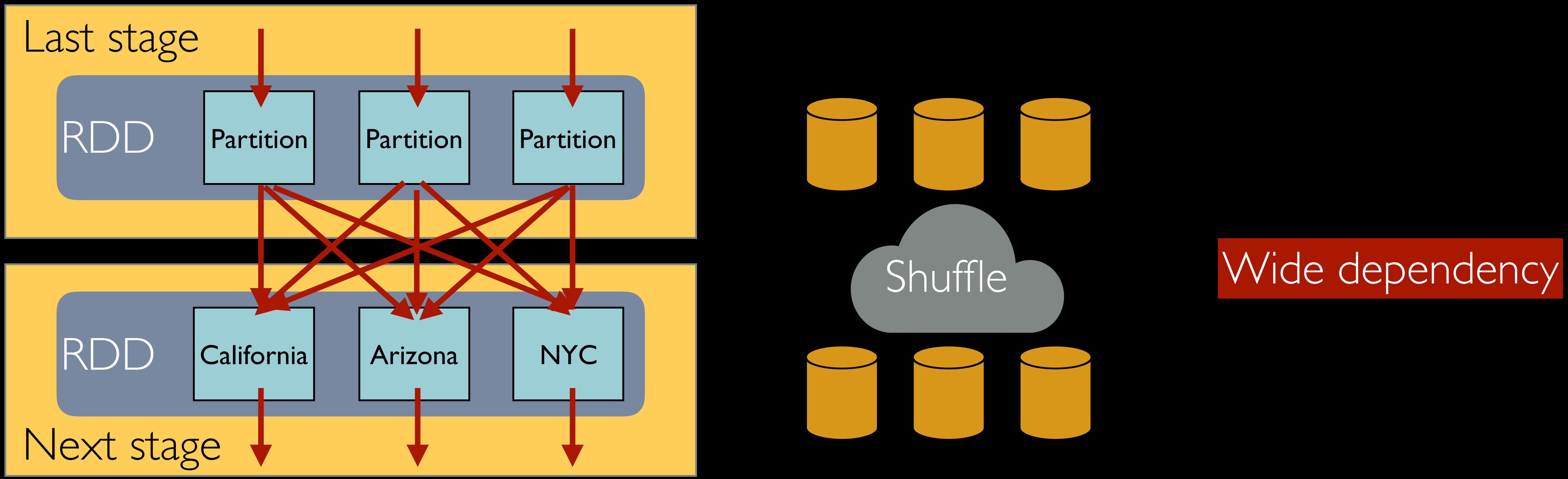




Yu, Jia, Zongsi Zhang, and Mohamed Sarwat. "Spatial data management in apache spark: the GeoSpark perspective and beyond." *GeoInformatica* (2018): 1-42.

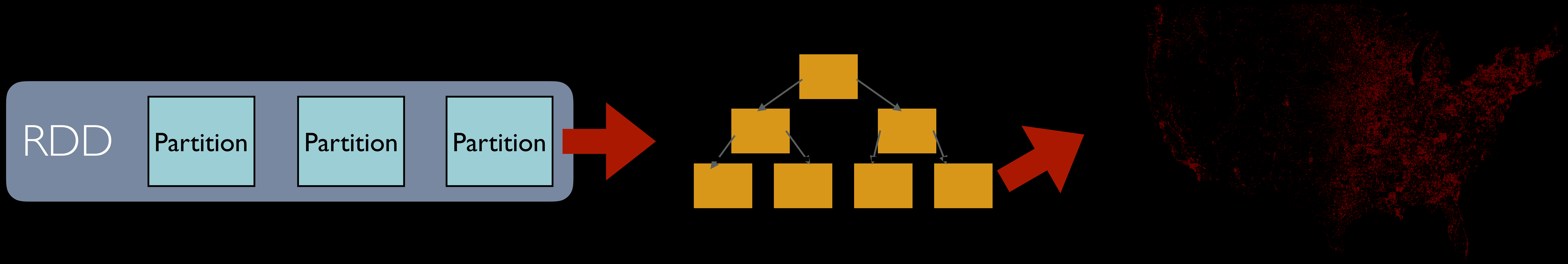
Spatial Data Partitioning

- Each spatial partitioning operation is a wide dependency
- Wide dependency will incur a data shuffle



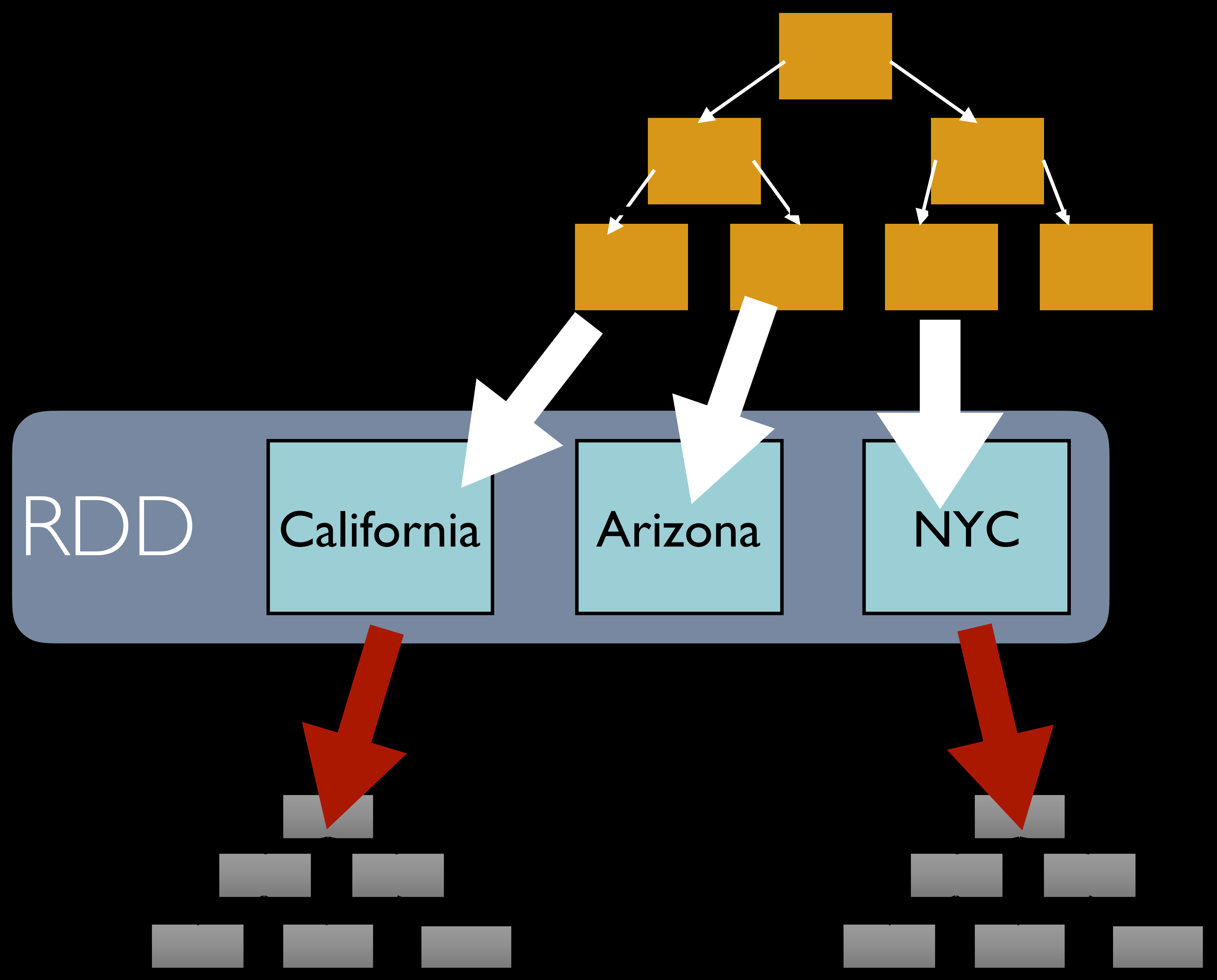
Spatial Data Partitioning

- Spatial partitioning algorithm
  - Randomly sample the RDD
  - Build a KD-Tree/Quad-Tree/R-Tree on the sample
  - Take the leaf nodes of the tree as the global partition file
  - Re-partition the RDD according to the partition file



Spatial Indexing

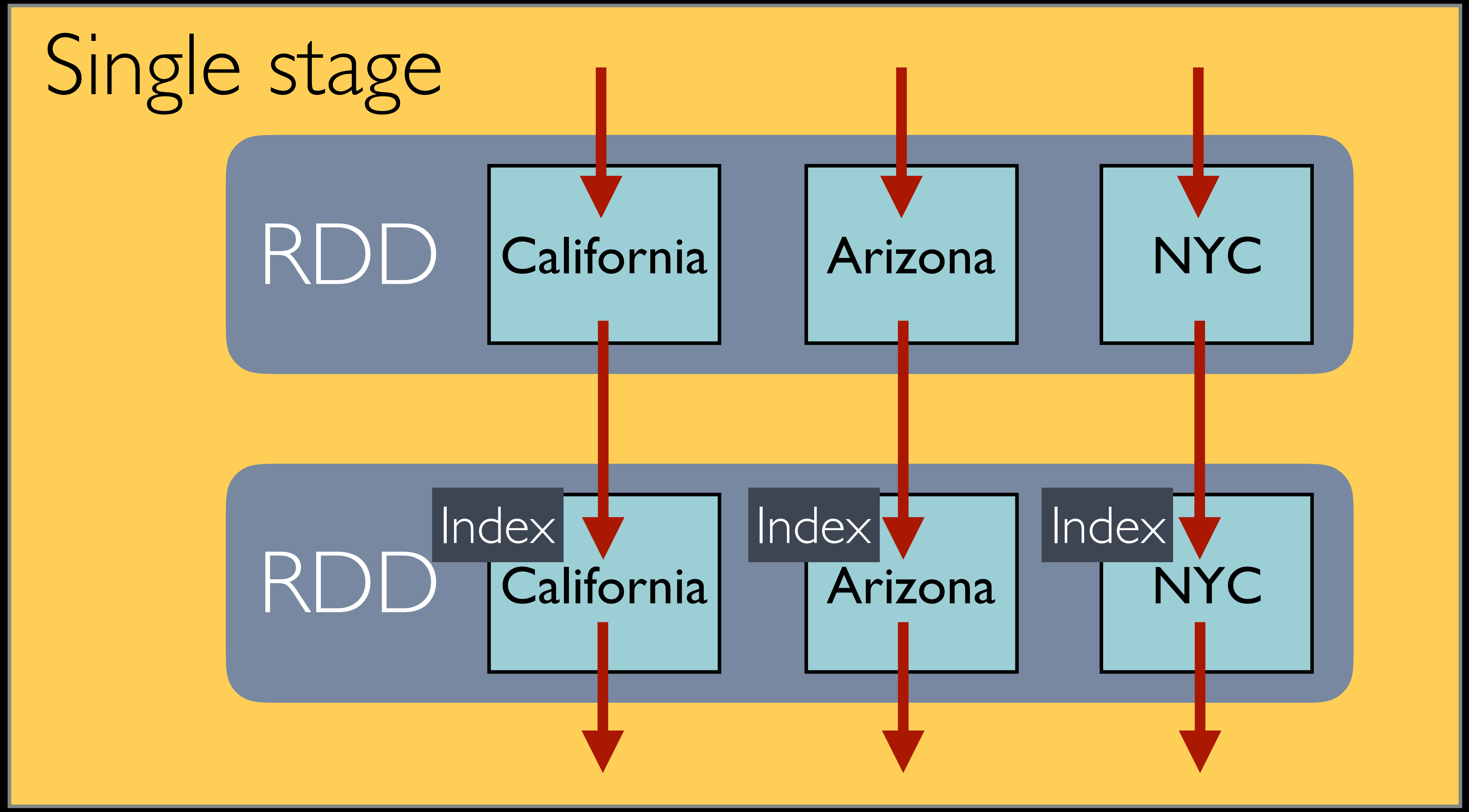
- Global index
  - Remember the tree built for spatial partitioning?
  - Use it to index partition bounding boxes
  - Lightweight, on the master machine
  - No entries for individual records

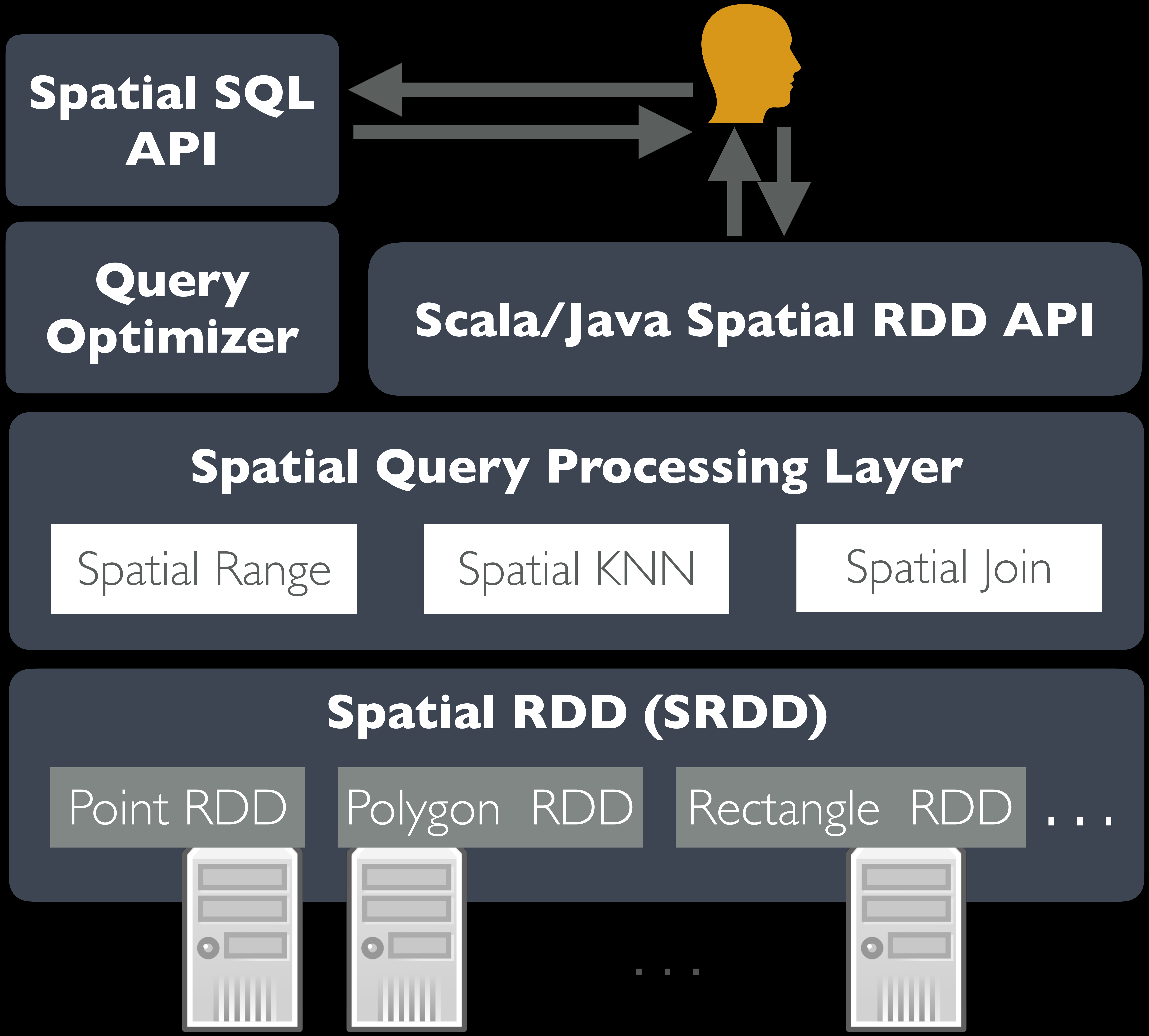


- Local indexing
  - On each SRDD partition
  - R-Tree, Quad-Tree,…
  - Has entries for individual records
  - Operations that use the spatial index <u>requires a refinement phase based on the real shapes of objects</u>

Spatial Indexing

- DAG and data shuffle: 1 RDD transformations
  - Global indexing: done with the spatial data partitioning (including partition range index)
  - Local indexing: Map per Partition, Narrow dependency



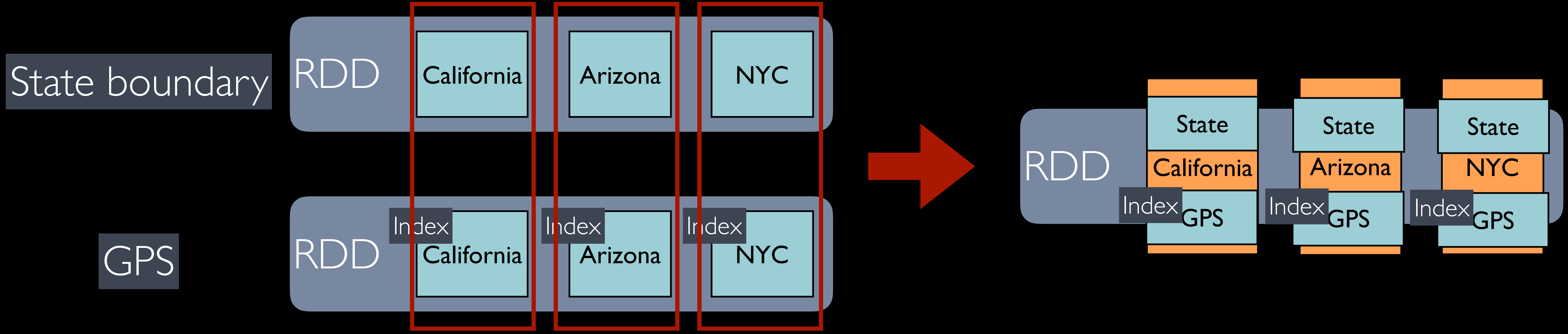

Spatial SQL API
Query Optimizer
Scala/Java Spatial RDD API
Spatial Query Processing Layer
Spatial Range
Spatial KNN
Spatial Join
Spatial RDD (SRDD)
Point RDD
Polygon RDD
Rectangle RDD
. . .
. . .

Spatial Join Query

- Algorithm
  - ZipPartition
  - Local index-nested loop join
  - Local de-duplication using the reference point

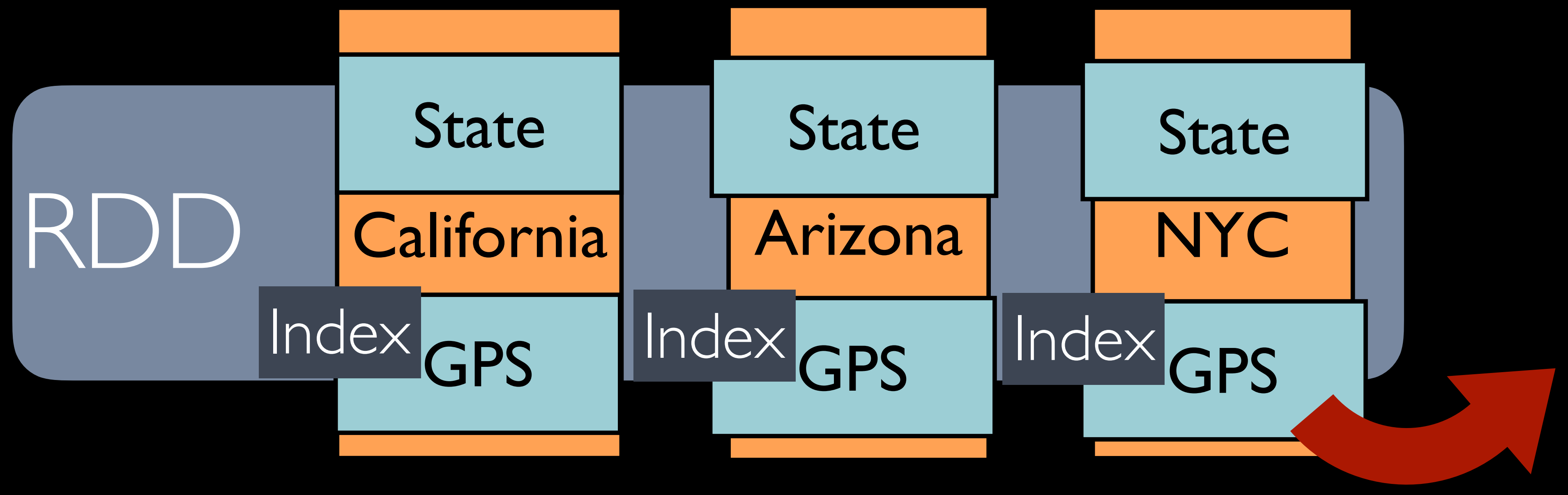
Spatial Join Algorithm

- ZipPartition
  - Both SRDDs should be co-partitioned
  - Any of the SRDDs can have local index



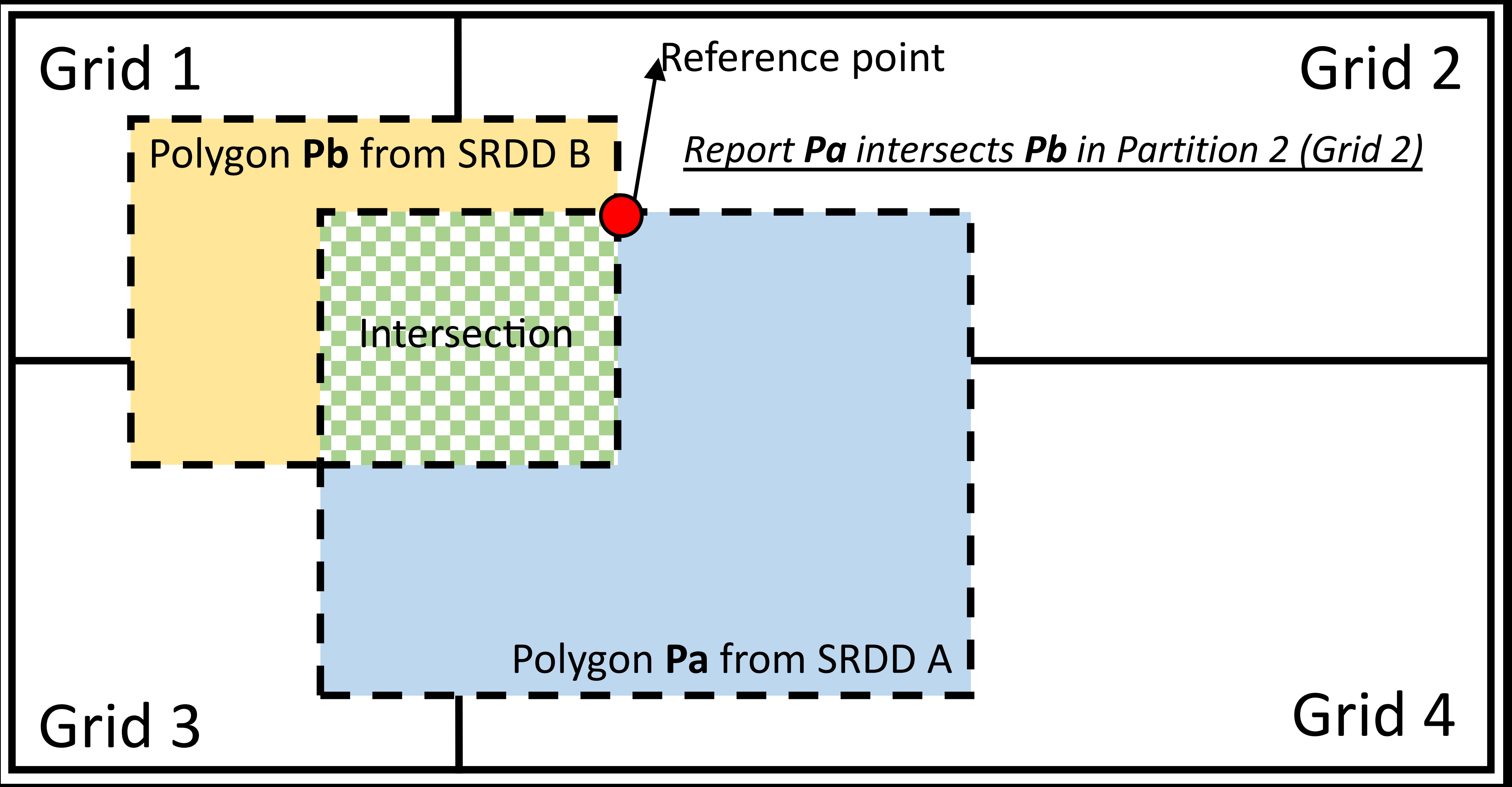
Spatial Join Algorithm

- Local index-nested loop join



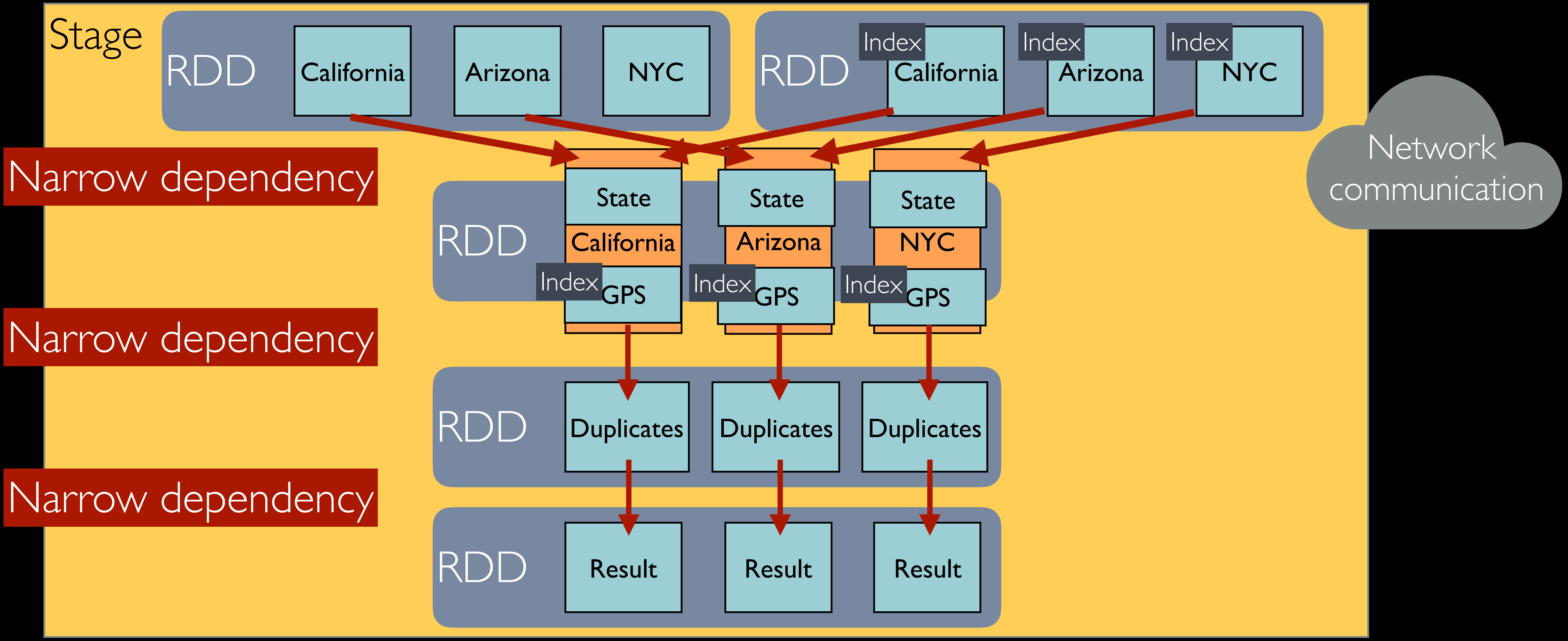
- Local de-duplication using the reference point
  - Spatial partitioning introduces duplicates
  - Need to remove them without incurring data shuffle!

Spatial Join Algorithm



- Reference point
  - Query results with duplicates
    - (Pa, Pb) (Pa, Pb) (Pa, Pb) (Pa, Pb)
  - Compute the intersection of Pa and Pb
  - Take Reference Point(maxX, maxY) of intersection
  - Report (Pa, Pb) in a partition only if reference point is within the boundary of this partition

Spatial Join

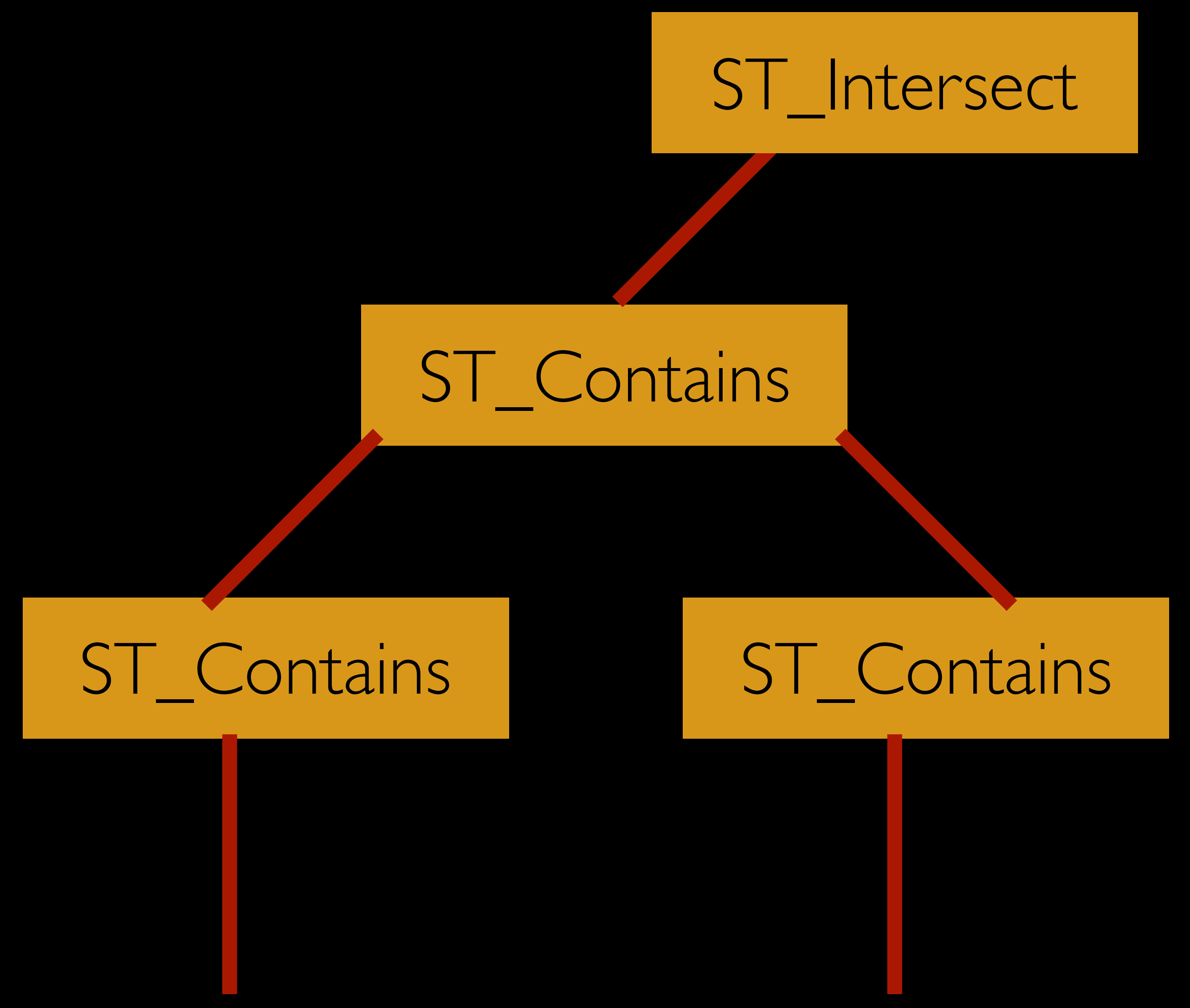


Integrate With Dataframe Spatial SQL

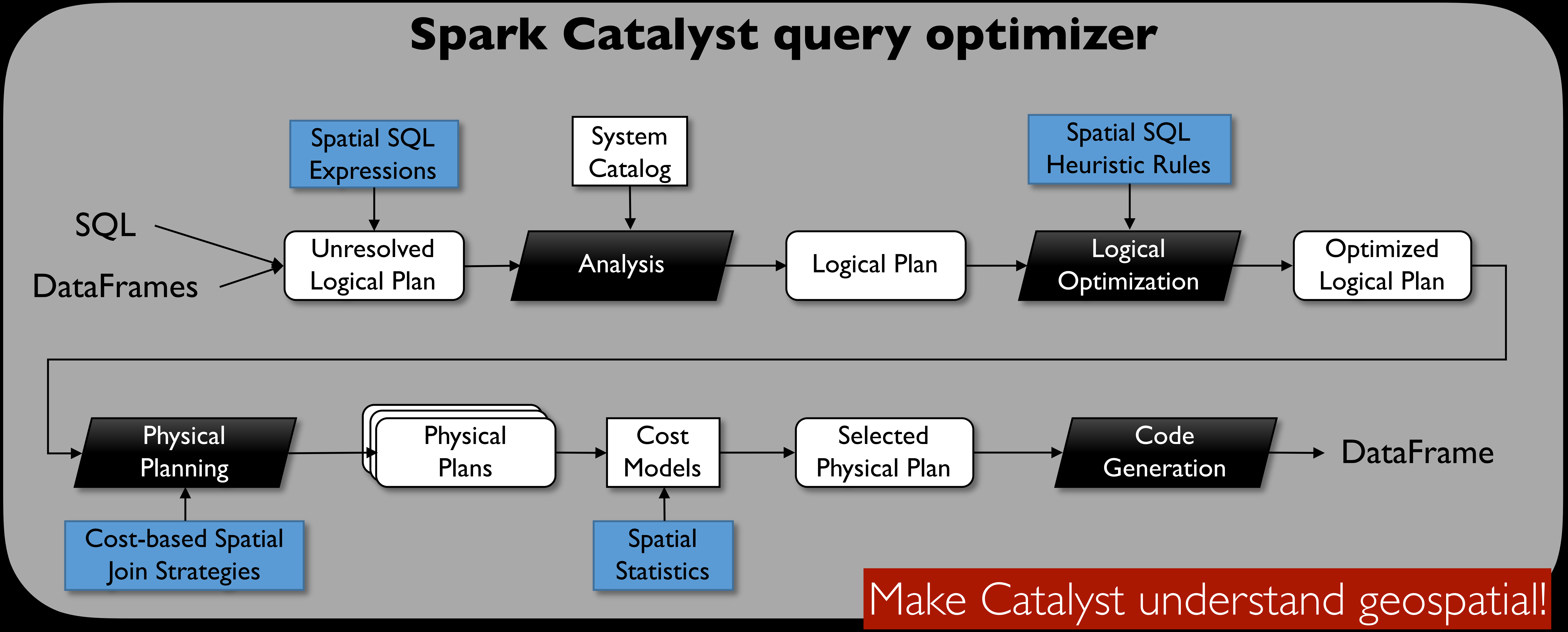
```
SELECT trip.* FROM   NYCtaxi trip, GooglePlaces place
WHERE  trip.NumPassengers = 1 AND place.type = ‘Hospital’
 AND   ST_DWITHIN(place.loc, trip.dropoff, 10)
```

- Spatial SQL: SQL-MM3, Simple Feature Access
  - SQL-MM3: PostGIS, GeoSpark, GeoMesa…
    - *ST_Contains, ST_Within*
  - Simple Feature Access
    - *Contains, Within*
  - Compatible with each other in most cases
  - Implement these functions in Spark expression (not UDF)

- Spark expression, the way Spark writes its own functions
  - Unary, binary, ternary
  - Each AST (Abstract Syntax Tree) node is a Spark expression
  - Allow the following features
    - Code generation
    - Output data type
    - Fuse into the Catalyst optimizer



Integrate With Dataframe



Integrate With Dataframe

GeoSpark performs **spatial join** on **1.7 billion** GPS data and **200 thousands** polygon data in ~**3 mins** on 4 commodity machines

```
:- Project [st_polygonfromenvelope(XXX), mypolygonid) AS polygonshape#20]
:  +- *FileScan csv
+- Project [st_point(XXX) AS pointshape#43]
   +- *FileScan csv
```

GeoSpark

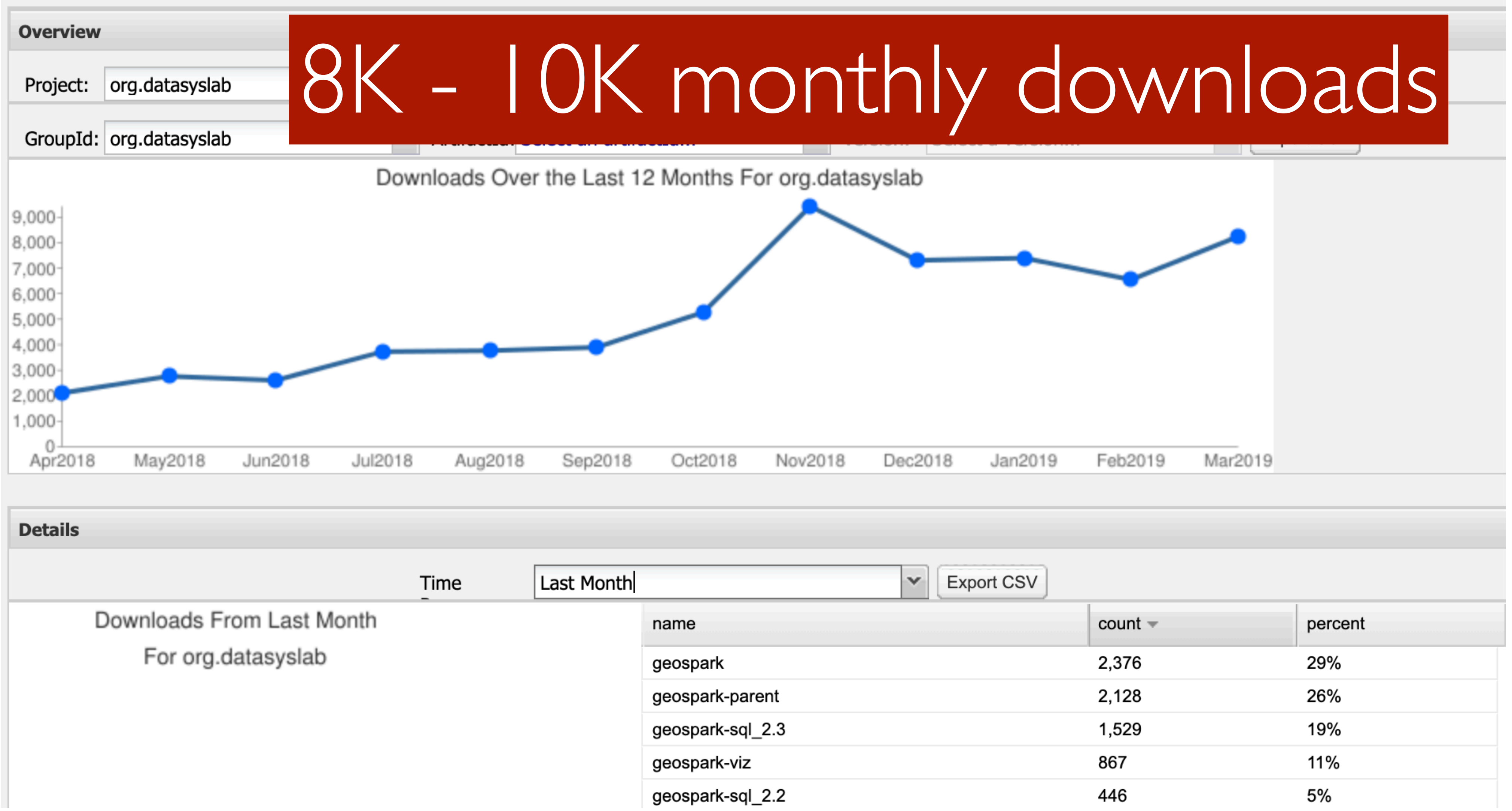
http://datasystemslab.github.io/GeoSpark/

Google "GeoSpark ASU"

- All-in-one system
- Spatial RDD, Spatial SQL, Spatial DataFrame
- Distributed map visualization is included
- Welcome to use GeoSpark as a benchmark!

In production!

8K - 10K monthly downloads



"GeoSpark comes close to a complete spatial analytics system. It also exhibits the best performance in most cases."

"How Good Are Modern Spatial Analytics Systems?" Varun Pandey, Andreas Kipf, Thomas Neumann, Alfons Kemper, PVLDB 2018

Scala API

Spatial SQL

GeoViz API

GeoSimulati on API

Temporal

Trajectories

GeoSpark

Geospatial Statistics

Spatial Streaming

ESRI ShapeFiles

GeoJSON Docs

NASA HDF/ NetCDF

PostGIS DB

esri

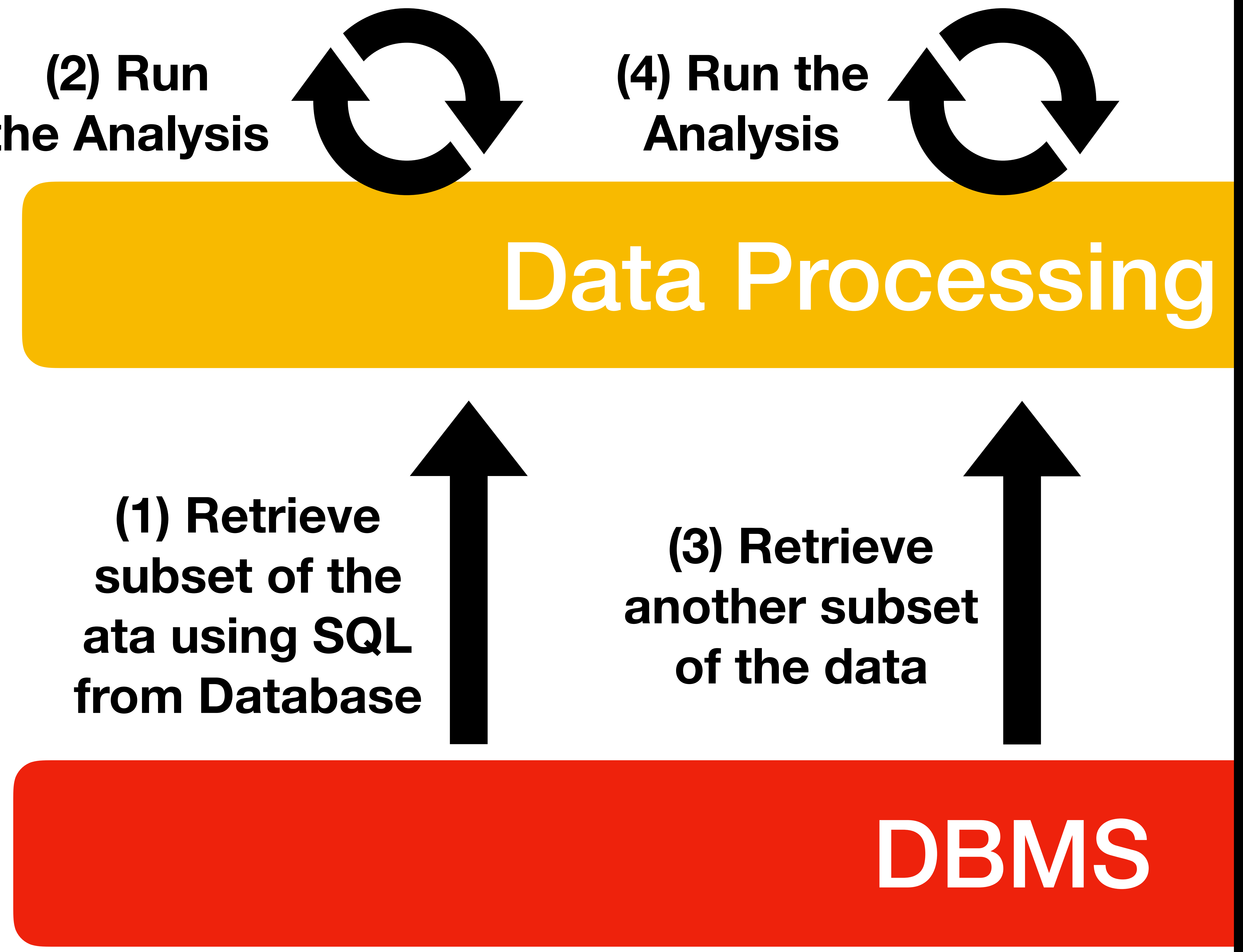
mongoDB

NASA

PostGIS

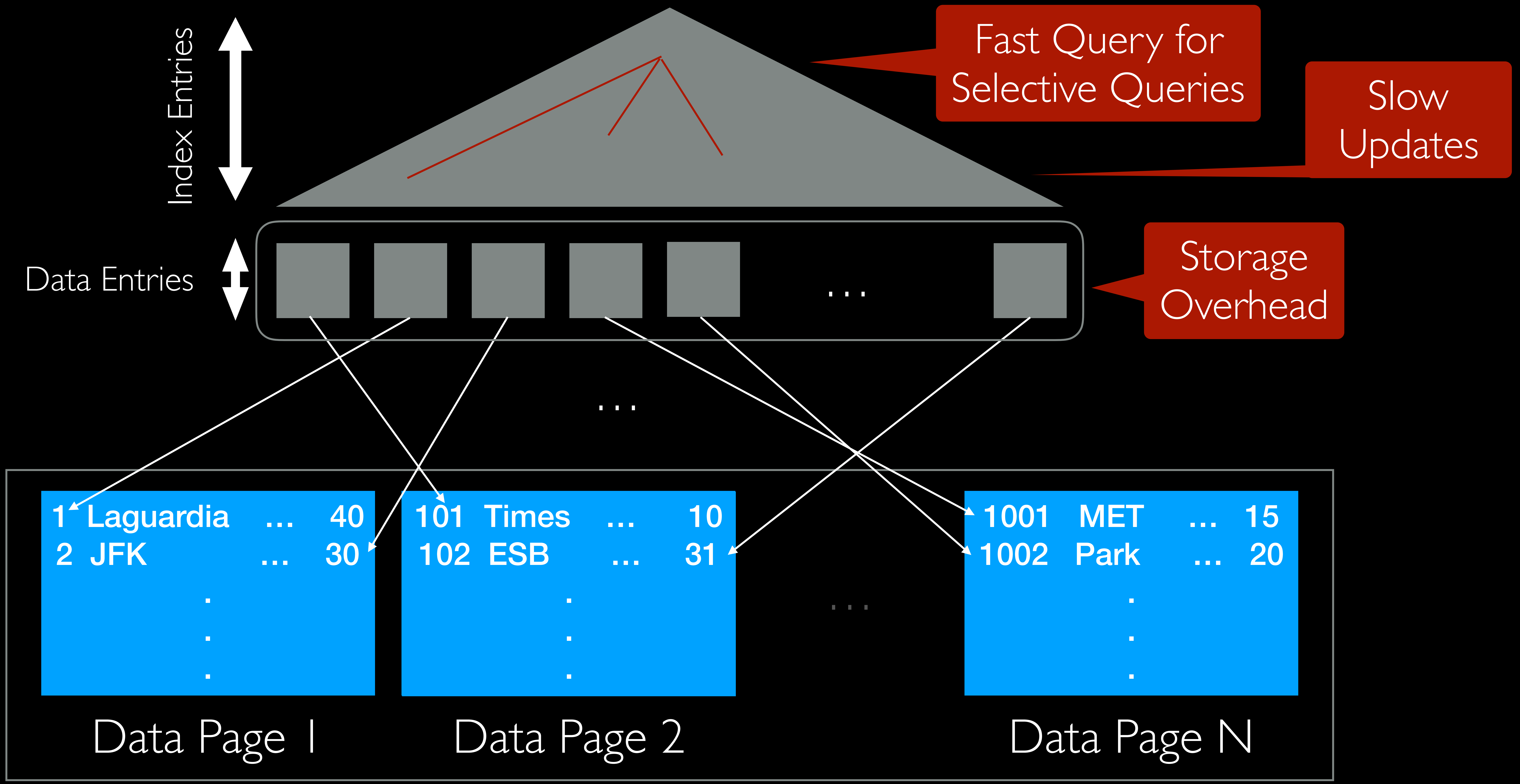
Spatial PostgreSQL

Jia Yu, Jinxuan Wu, ,M. Sarwat, "A demonstration of GeoSpark: A cluster computing framework for processing big spatial data" in IEEE ICDE 2016



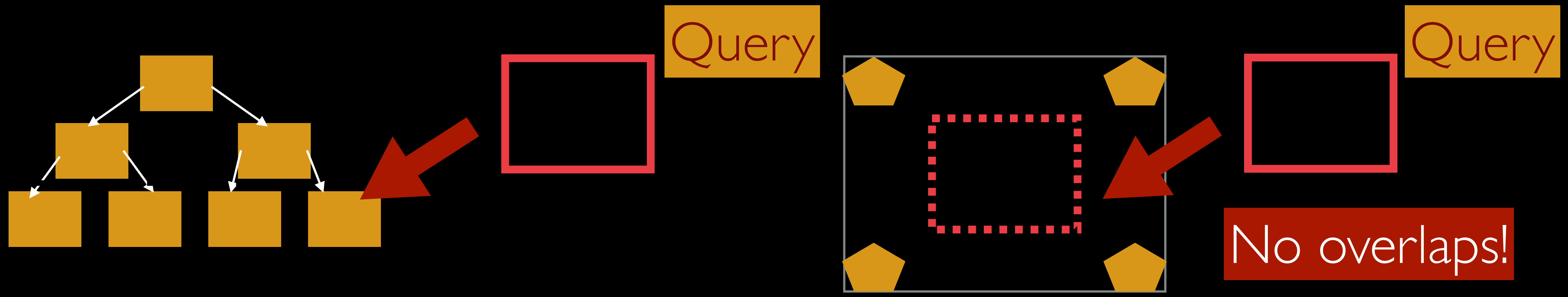
- **Lightweight index**
  - Small storage footprint
  - Low maintenance overhead
- **Comparable Query Performance**
  - nearly as fast as the B-tree, R-tree, especially for selective queries (0.1%< SF <1%)

Creating a Secondary Index



New Indexing Scheme

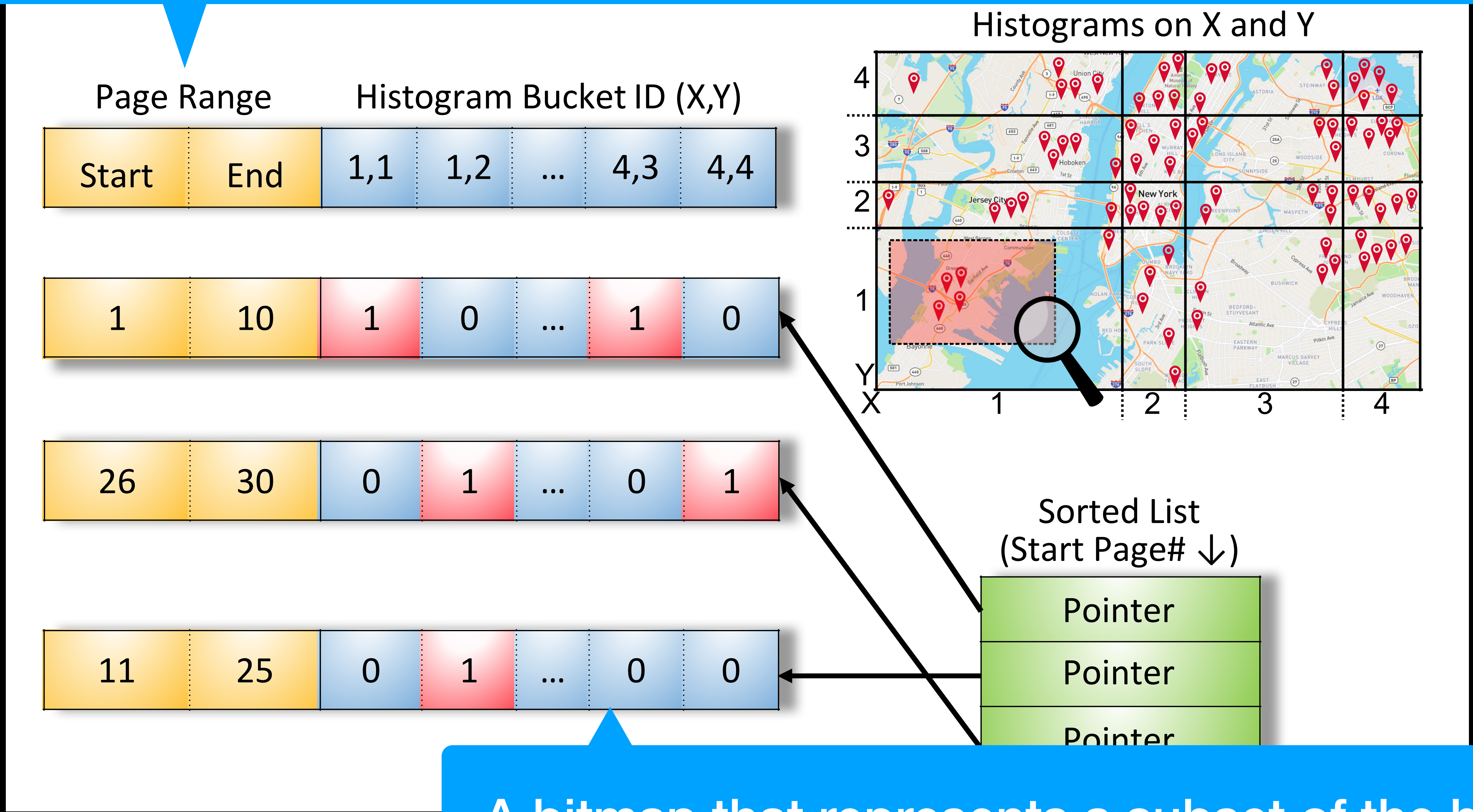
- The index only indexes blocks not data items, which significantly reduces the storage and maintenance overhead
- Queries sometimes still go to false positive blocks, which is not good for very highly selective queries but should ok for queries with medium selectivity factor



Jia Yu, M. Sarwat, "Tow Birds, One Stone: A Fast, yet Lightweight, Indexing Scheme for Modern Database Systems" in PVLDB 2016

Data-Aware Block Range Index

- The IDs of the first and last pages summarized by the index entry.
- Summarizes more than one physically contiguous pages to reduce the overall index size.



BlockID	Bucket(1,1)	Bucket(1,2)	Bucket(1,3)	...
0	1	0	0	
1	0	1	0	
2	0	1	0	
3	0	0	1	

- A bitmap that represents a subset of the histogram buckets.
- Each partial histogram represents the spatial data distribution of the summarized pages.

Yu, Jia, and Mohamed Sarwat. "Indexing the Pickup and Drop-Off Locations of NYC Taxi Trips in PostgreSQL–Lessons from the Road." In SSTD 2017

Hippo

FAST, YET LIGHTWEIGHT, INDEXING SCHEME

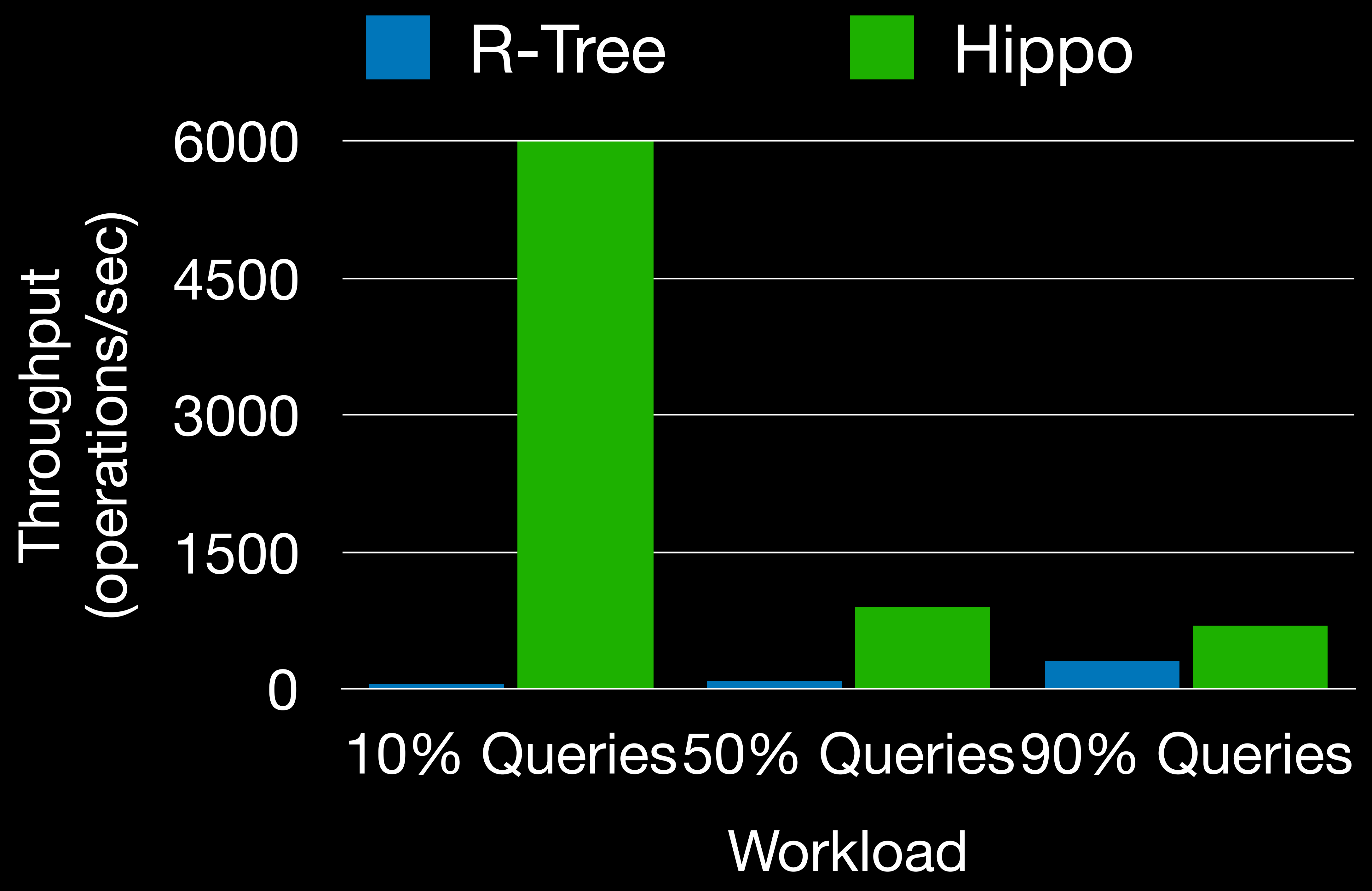
https://github.com/DataSystemsLab/hippo-postgresql

Jia Yu, Raha Murafah, ,M. Sarwat, "Hippo in Action: Indexing 1 Billion NYC taxi trips and Beyond" at IEEE ICDE 2017

	Our Solution	B+Tree	R-Tree
Storage	~1GB	~40 GB	~80 GB
Init.	~2700 sec	~7900 sec	~70,000 sec

INDEXING OVERHEAD

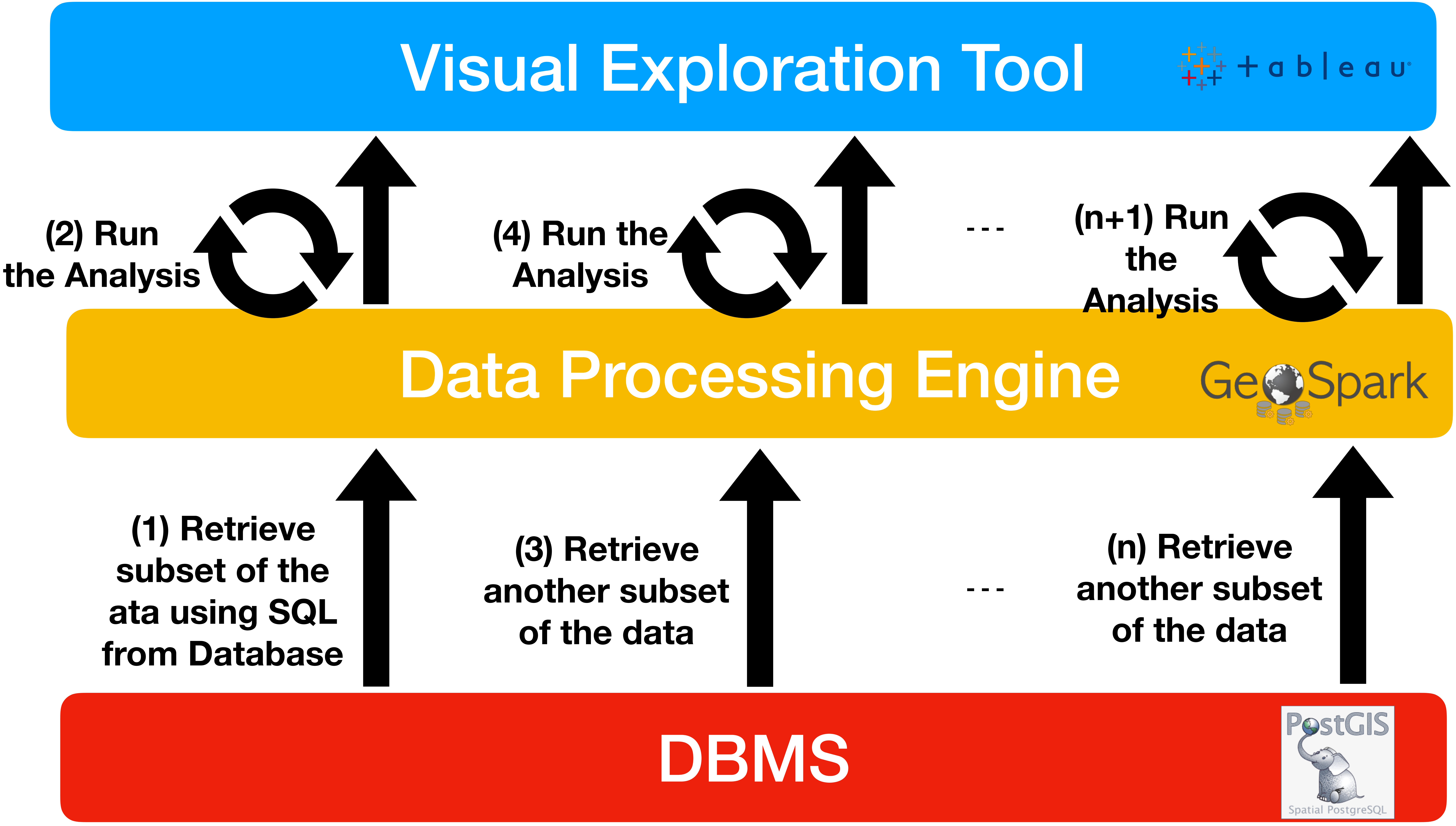
Hippo achieves about 40X less storage and about 60% less initialization time than the B+Tree



THROUGHPUT

Hippo achieves ~100x higher throughput for update intensive workloads and ~10x for balanced workload

PostgreSQL currently implements a similar idea


Visual Exploration Tool
tableau
(2) Run the Analysis
(4) Run the Analysis
- - -
(n+1) Run the Analysis
Data Processing Engine
GeoSpark
(1) Retrieve subset of the ata using SQL from Database
(3) Retrieve another subset of the data
- - -
(n) Retrieve another subset of the data
DBMS
PostGIS
Spatial PostgreSQL

Challenges

- Scalability
  - How to render a high quality gigapixel heat map or a dot map for Terabytes (or more) of geolocation data?
  - Massively parallelize the map rendering algorithm using GeoSpark ([SSDBM 2018], [ICDE 2019])
- Interactivity
  - How can a human seamlessly interact and visually explore with geospatial data
  - Materialization and Sampling

Switching Gears A Bit…

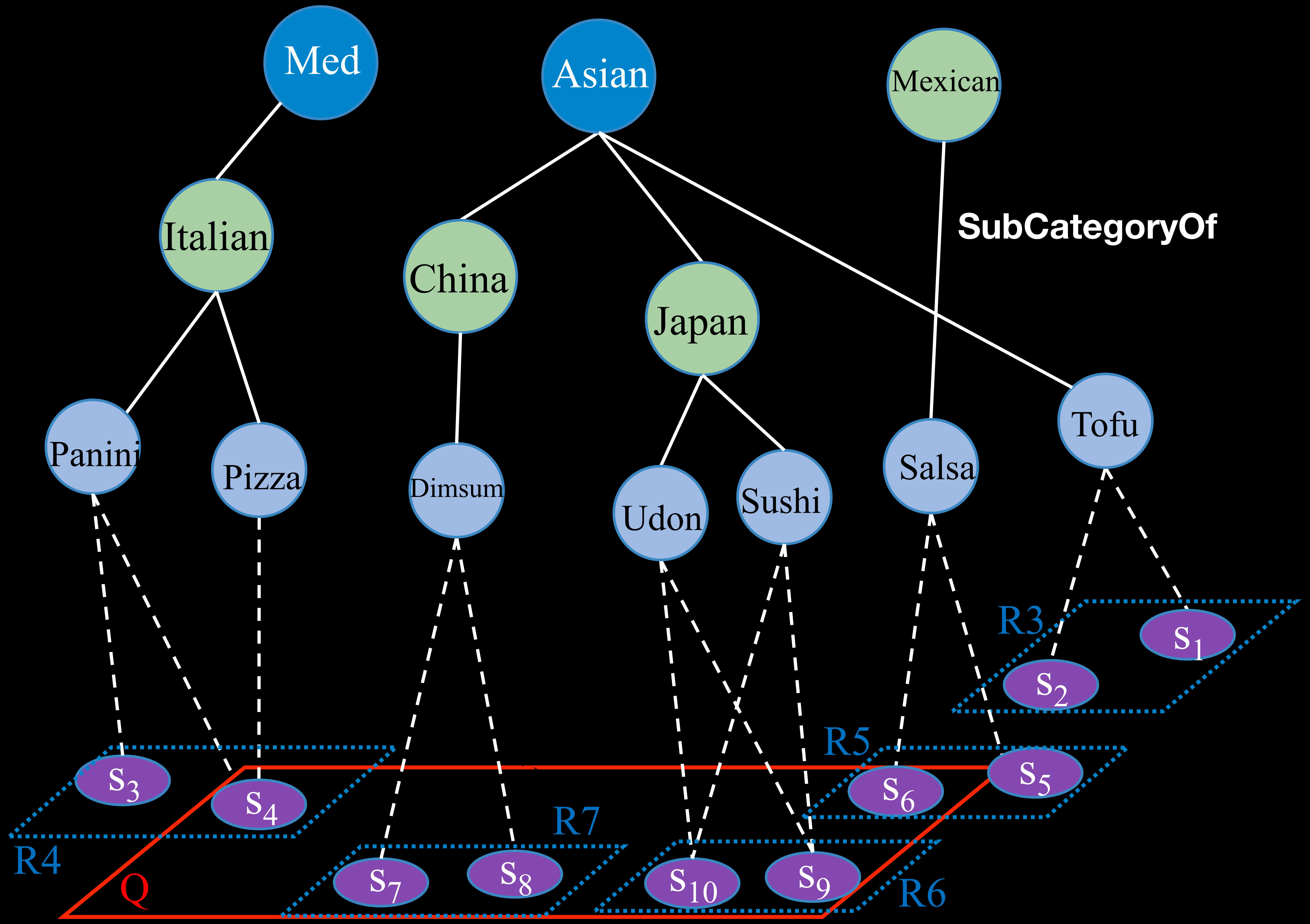
Graph Data

- Google Knowledge Graph
- Wikidata Knowledge Graph
- Semantic Web
- Social Graph
- …

12% to 30% of entities in such graphs are geospatial in nature

UberEats Knowledge Graph

- Assume the following graph is stored in Neo4j
- Find Mediterrenean restaurants within the Mong Kok area
- Find Asian cosines nearby the HKBU campus



Two Solutions

Graph Data System

- Neo4j
- Titan
- Apache TinkerTop
- …

VS

Spatial Data System

- PostGIS
- Oracle Spatial
- GeoSpark
- …

Graph DBMS & Spatial DBMS

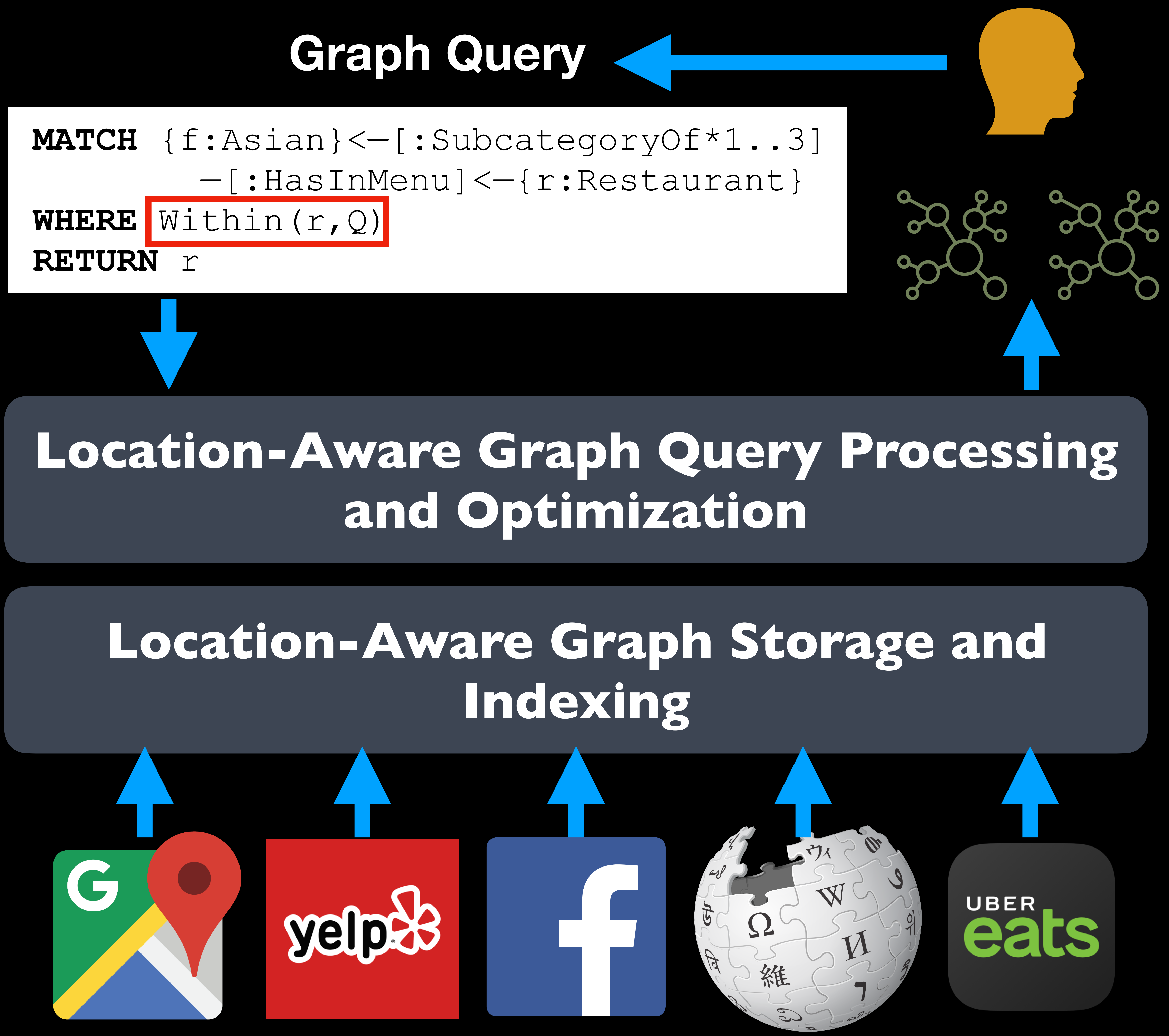
- **Spatial DBMS**
  - Provides a geospatial data processing API
  - Does not provide a Graph data processing API
  - Can Efficient optimize geospatial predicates

```
MATCH {f:Asian}<—[:SubcategoryOf*1..3]
        —[:HasInMenu]<—{r:Restaurant}
WHERE Within(r,Q)
RETURN r
```

- **Graph DBMS**
  - Provides a graph processing API
  - Provided a geospatial data processing API
  - Could NOT efficiently optimize geospatial predicates

SPINDRA

- The system stores data natively as a graph in Neo4j (the same applies to other graph database systems)
- The system takes a graph query as input from the user and then returns the graph pattern(s) or paths (s) that match the user query
- The queries can involve spatial predicates like Range, KNN or Spatial Join…



SPINDRA

[IEEE ICDE 2019] [GeoInformatica 2019]
[ACM SIGSPATIAL 2018]

Graph Query

```
MATCH  {f:Asian}<—[:SubcategoryOf*1..3]
              —[:HasInMenu]<—{r:Restaurant}
WHERE  Within(r,Q)
RETURN  r
```

Range+G | **KNN+G** | **Join+G**

Location-Aware Graph Traversal

Location-Aware Graph Traversal Operator

Spatial Properties in Graph Vertex and Edges

Location-Aware Graph Query Processing and Optimization

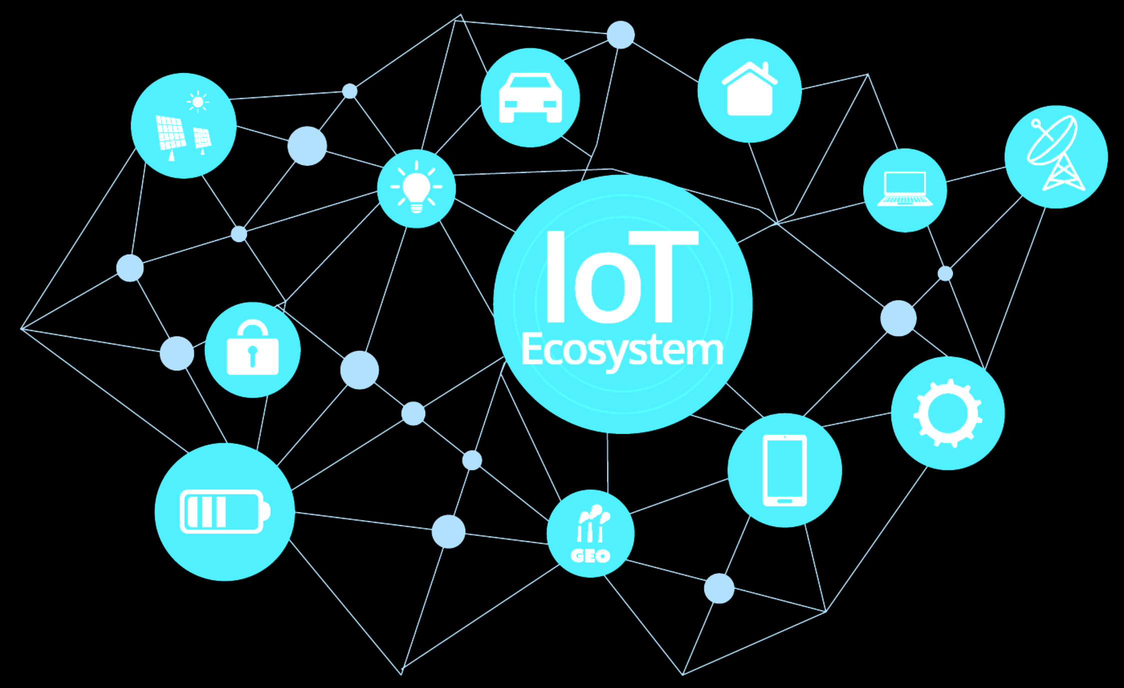
Location-Aware Graph Storage and Indexing

Future Work

- Building a Comprehensive Geolocation Knowledge Graph
  - A repository for facts (and events) about (at) geospatial locations
  - Challenge: spatial data integration, spatial entity linkage/resolution
- A Conversational interface to Spatial Databases
  - While driving you say "Hey Google, What are good restaurants for dinner"
  - Challenge: Natural Language to Spatial Queries and Understanding/Refning the human intent
- Spatial Data Systems support for the Internet of Things (IoT)

The Internet of Things

- Things (non-traditional computing devices) that can send / receive data via the Internet.



- Opportunities:
  - Every piece of data has a geospatial location and a time stamp
  - Some of these things are mobile (e.g., Smart car)
  - 5G will tremendously increase the scale of such data

IoT - Data Management Challenges

- **Heterogeneity on steroids**
  - IoT devices have totally different HW and SW.
  - Each device generate different physical measurements (e.g., Temperature, LiDAR, Humidity, GPS,
  - Each device sends data back to the central data system at different rates
- **A need for ... together:**
  - Signal Processing
  - Machine Learning
  - Spatial and SpatioTemporal Data management
- **Privacy?**

All of that needs to work in real-time for data being inserted at a staggering rate

Some Lessons from the Road

Myths About Geospatial Data

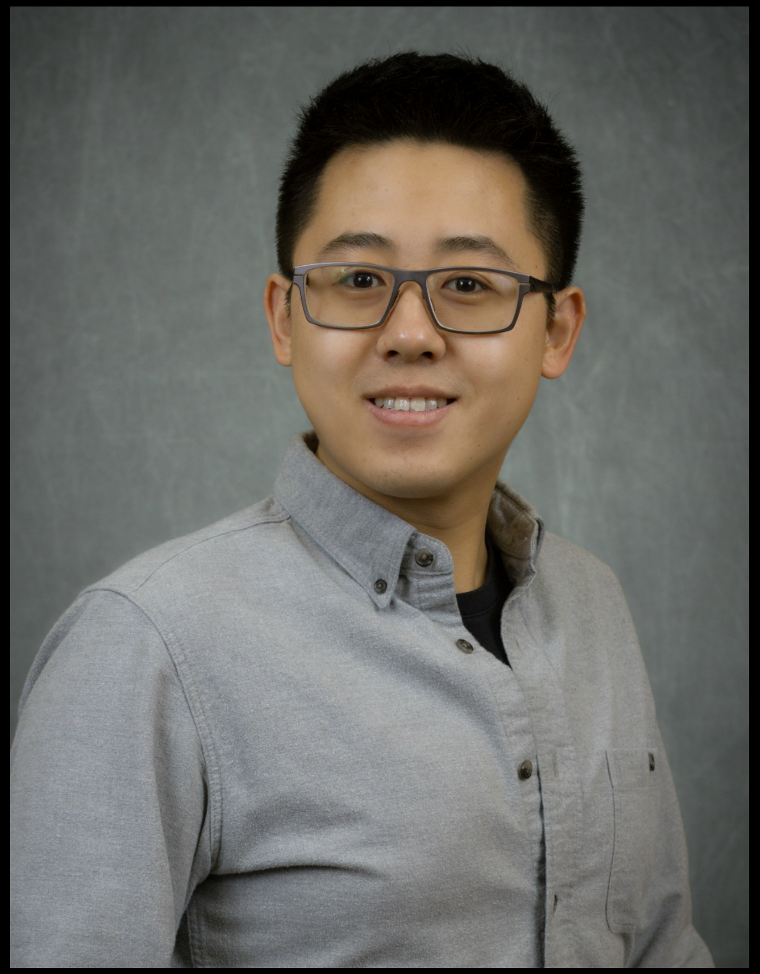
- Myth 1: Geolocation is yet another attribute
- Myth 2: Geolocation is the most important attribute
- Myth 3: All existing data systems cannot handle Geolocation data

What To Optimize a System for?

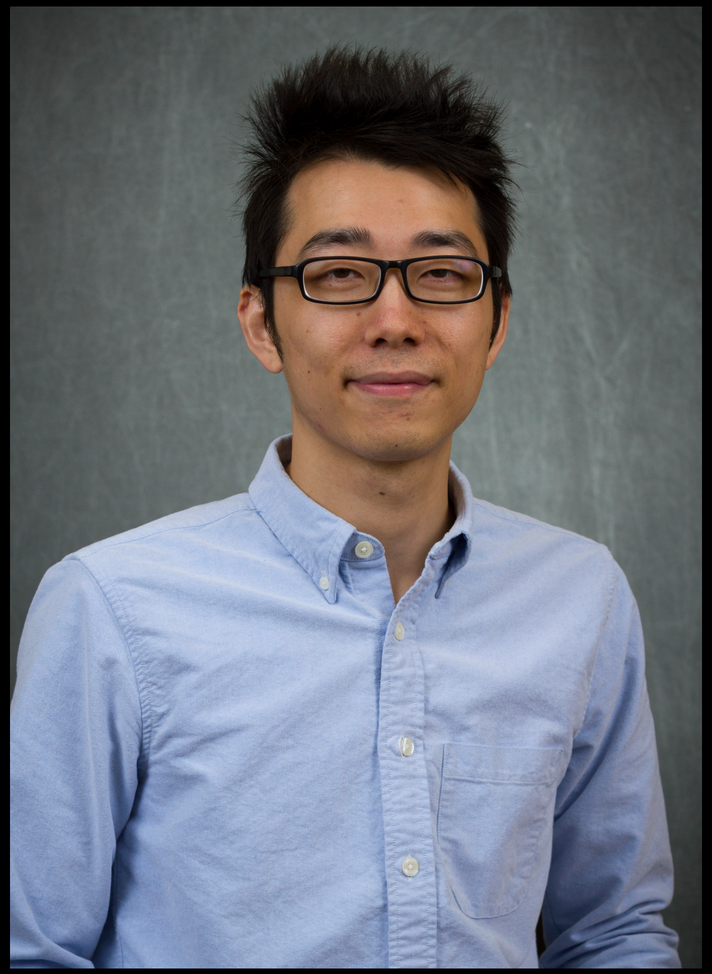
- Usability
- Interoperability
- Modularity
- Interactivity
- Scalability

Acknowledgment

PhD Students



Jia Yu



Yuhan Sun

Vamsi Meduri

Kanchan Chowdhury

Sponsors

NSF

SRP

U.S.ARMY

Tempe

DataSys

https://www.datasyslab.net

@DataSysLab

@MoSarwat