

Balanced Geohash Partitioning and Efficient Retrieval of Geospatial Big Data on Distributed and Parallel Platforms

(Apache SparkTM)



[Tweet](#)

Hariharan
@hariharan

#RefugeeBoat
a degree online
#CodeFuture

Heike van Geel

The interface consists of three main vertical panels. The left panel shows a weather forecast for Darmstadt, Germany, with a large '18°' indicating temperature and a map below it. The middle panel is a map from RMV showing a street view with buildings numbered 43, 45, 10, 8, 6, 4, 51, 53, 124, 126, 130, 132, 127, 129, 131, 125, 125b, 179, and 173. A green location marker with a white 'H' is placed near the center. Below the map are buttons for 'Bahnhof / Haltestelle' and 'Darmstadt Hochschule'. The right panel is a 'Planner' section from Google Maps, showing a route from 'Darmstadt Hbf' to 'Frankfurt(Main)Hbf'. It includes a search bar, departure date and time ('Today, We, 13.07. | Departure 09:32'), and a red 'Search' button. At the bottom, there are navigation icons for 'Buy a ticket', 'Tickets', 'Favourites', 'Timetables', and 'More'.

Darmstadt, Germany

TODAY

July 12, 23:18

Day 24°↑ • Night 13°↓

18°
Feels like 18°

Burschenschaft Rheno-Markomannia

Studentenwohnheim

Mathilde Mode und Dekoration

Bahnhof / Haltestelle

Darmstadt Hochschule

from
Darmstadt Hbf

to
Frankfurt(Main)Hbf

Today, We,
13.07.

Departure
09:32

2 RETWEETS 1 LIKES

Reply to Hariharan

Add this location to your list

RMV

← 🔍 ⋮

☰ Darmstadt, Germany

📍

☰

Planner

Google

Traubenberg

Darmstadt Hbf S

Goebelstraße

from
Darmstadt Hbf

to
Frankfurt(Main)Hbf

Today, We,
13.07.

Departure
09:32

2nd Class | 1 Ad | BC 50 2nd Cl.

Search

Als Start übernehmen

Als Ziel übernehmen

Abfahrt / Ankunft

€ Buy a ticket

Tickets

★ Favourites

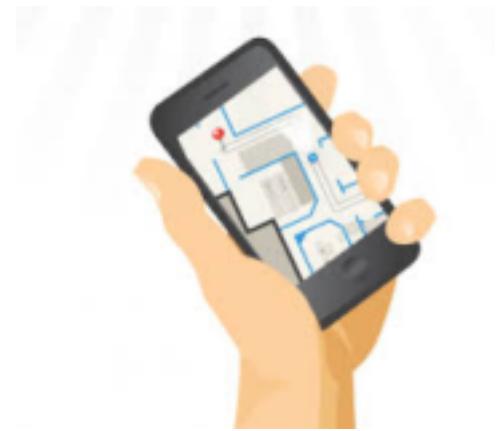
Timetables

More

Location based service - users (millions)



Location based market spending (billions)



7.6 Billion / 4.7 Billion unique - 2015

Trending Scenario

- ❖ Very large volume of Geo-spatial data
- ❖ Businesses need location aware services like,
 - tweets in an particular area
 - weather information of a holiday spot which you plan to visit over the vacation
 - Attractions nearby
 - Take me through the shortest route
 - many more.....

Seek support from Parallel computing platforms

- ❖ To handle such large volumes
- ❖ But unfortunately, **no native support** (yet)
- ❖ Needs modifying internal implementations



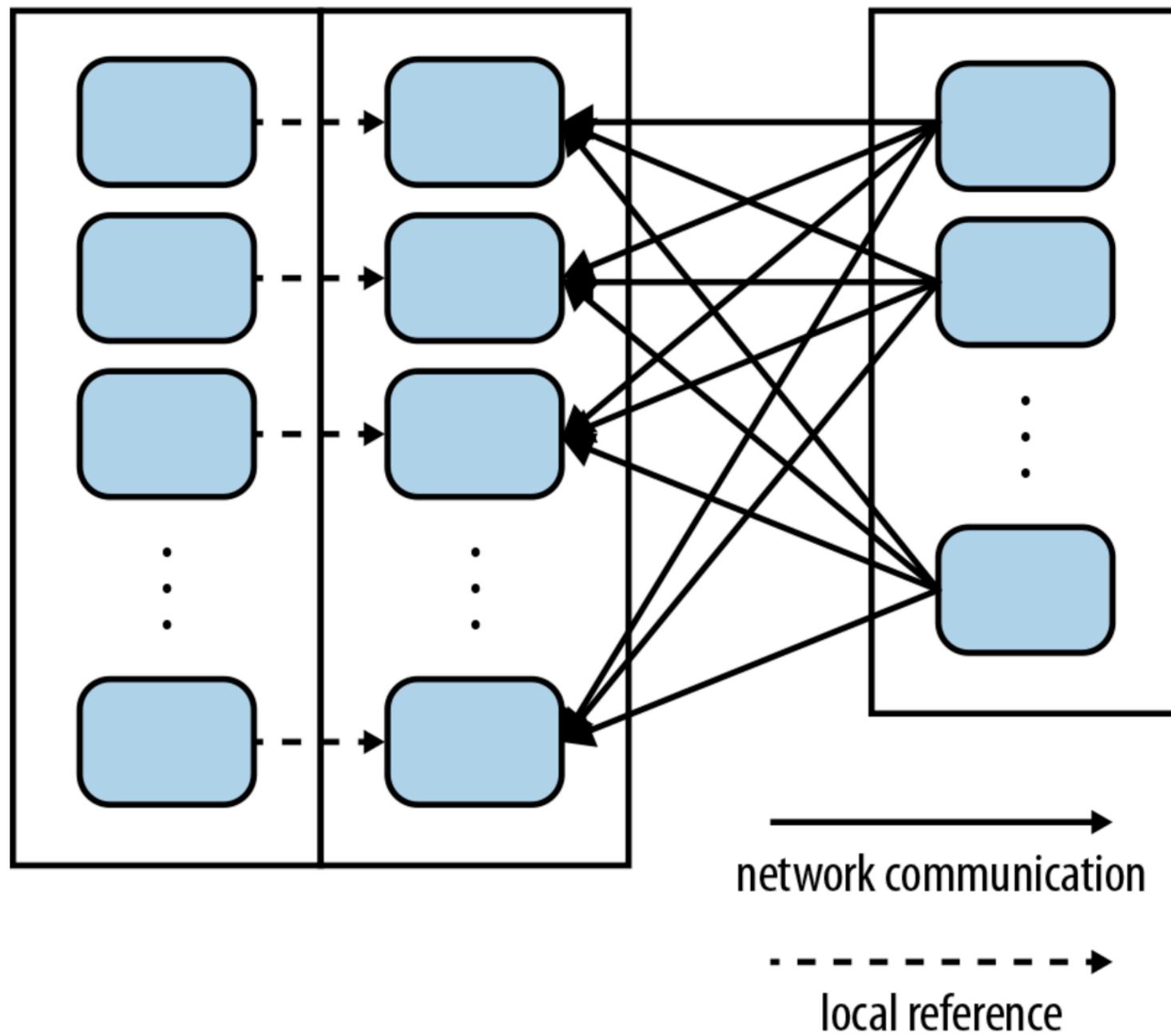
Why partition ?

- ❖ Communication in distributed platform is very **expensive**
- ❖ Range Queries, grouping – **Network cost** due to shuffling between clusters/nodes
- ❖ Distribute data to minimize network traffic → **performance**
- ❖ Single node -→ Distributed nodes

Large RDD

Join()

Small RDD

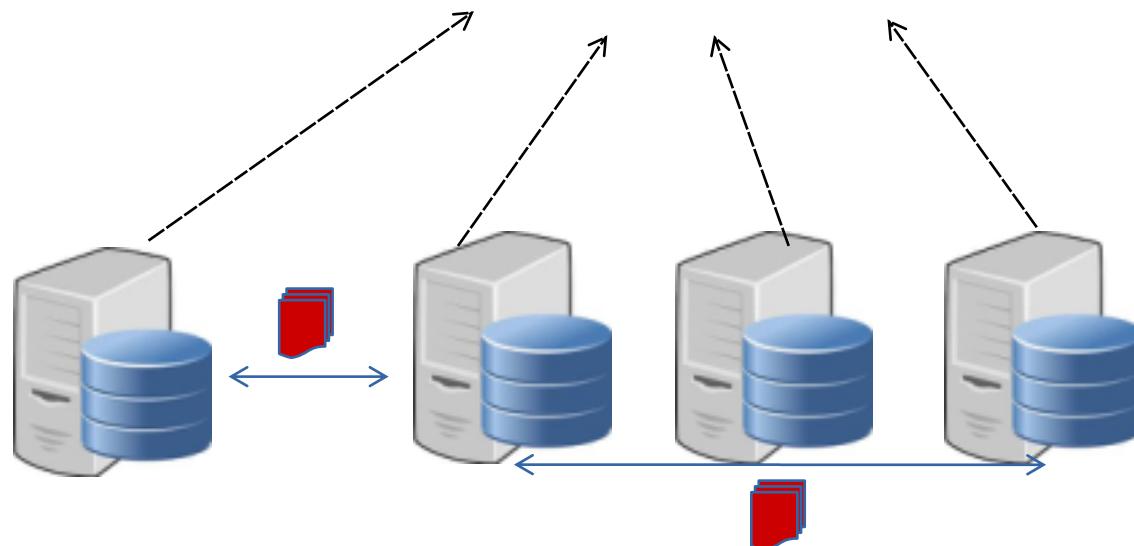


Problem is clear

For a query like,

"Give me all tweets activities that happened around my campus (say within 300 meter)"

- ❖ Tweets with geo tag TUDarmstadt is distributed across all partitions in the cluster
- ❖ Shuffles them between partitions (and in turn between nodes)
- ❖ Scans a larger section(say, records of whole Germany)



When to partition ?

- ❖ Geospatial operations intense application,. Eg: Location aware services
- ❖ Major interests of queries are Geospatial:
 - ❖ Ex: **Maps, Uber, Jodel, AirBnB, Emergency Services**
- ❖ Multiple reads

The plan

Introduce a partitioning scheme in Apache Spark that partitions data-sets considering Geo-spatial info:

- ❖ **Communication in distributed environment:**
 - ❖ minimize network traffic by allowing spark program to control RDD partitioning
- ❖ **Retained Spatial Locality:**
 - ❖ nearby points are probably in the same partition / cluster
- ❖ Minimize data points shuffling during Geo-spatial queries
- ❖ Optimized for Search & Range queries on Geo-points

Efficient Spatial processing with Spark



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ❖ GeoHash Partitioner
- ❖ GeoHash Query Layer



Space Filling Curves

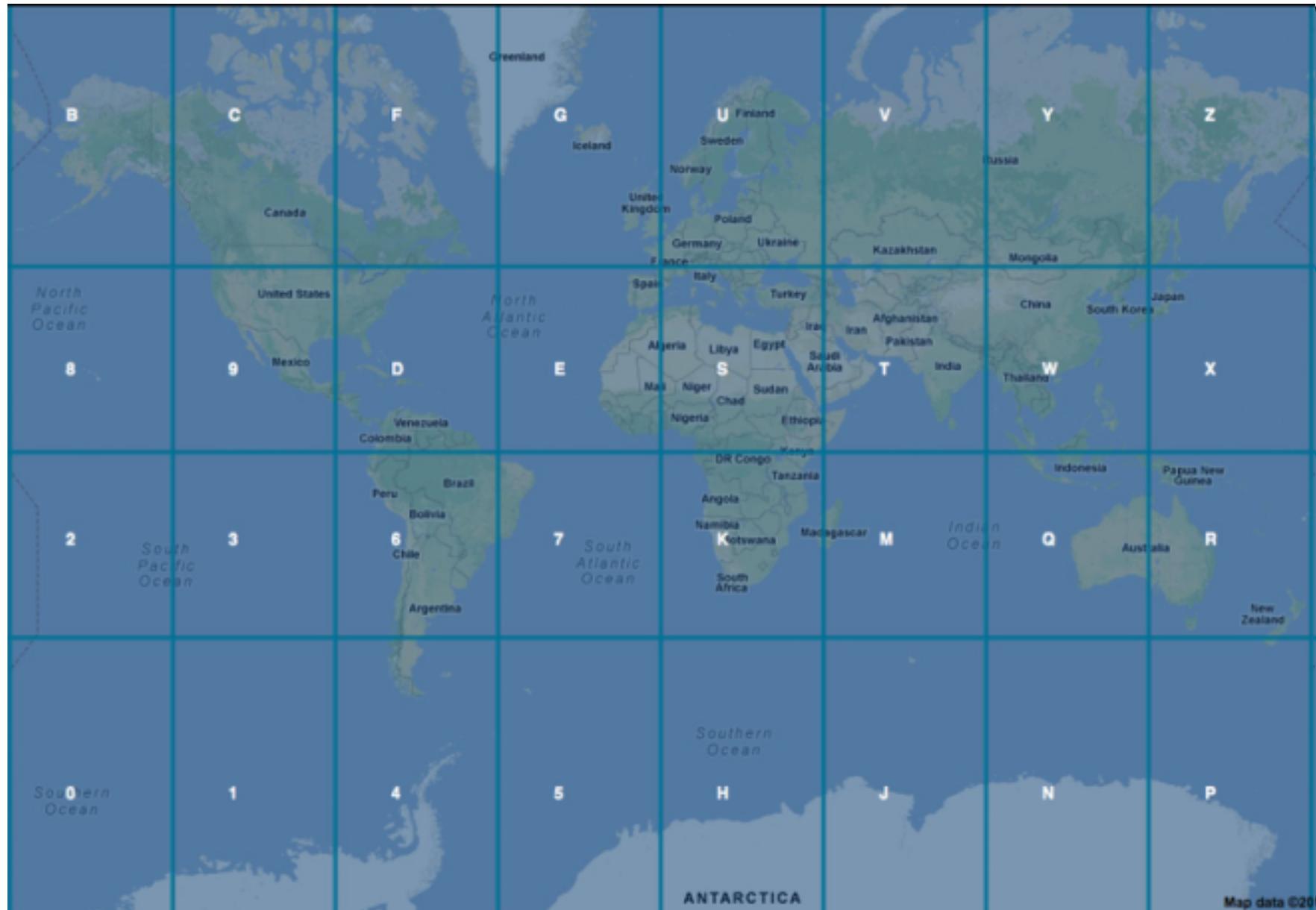
Z-order Curve



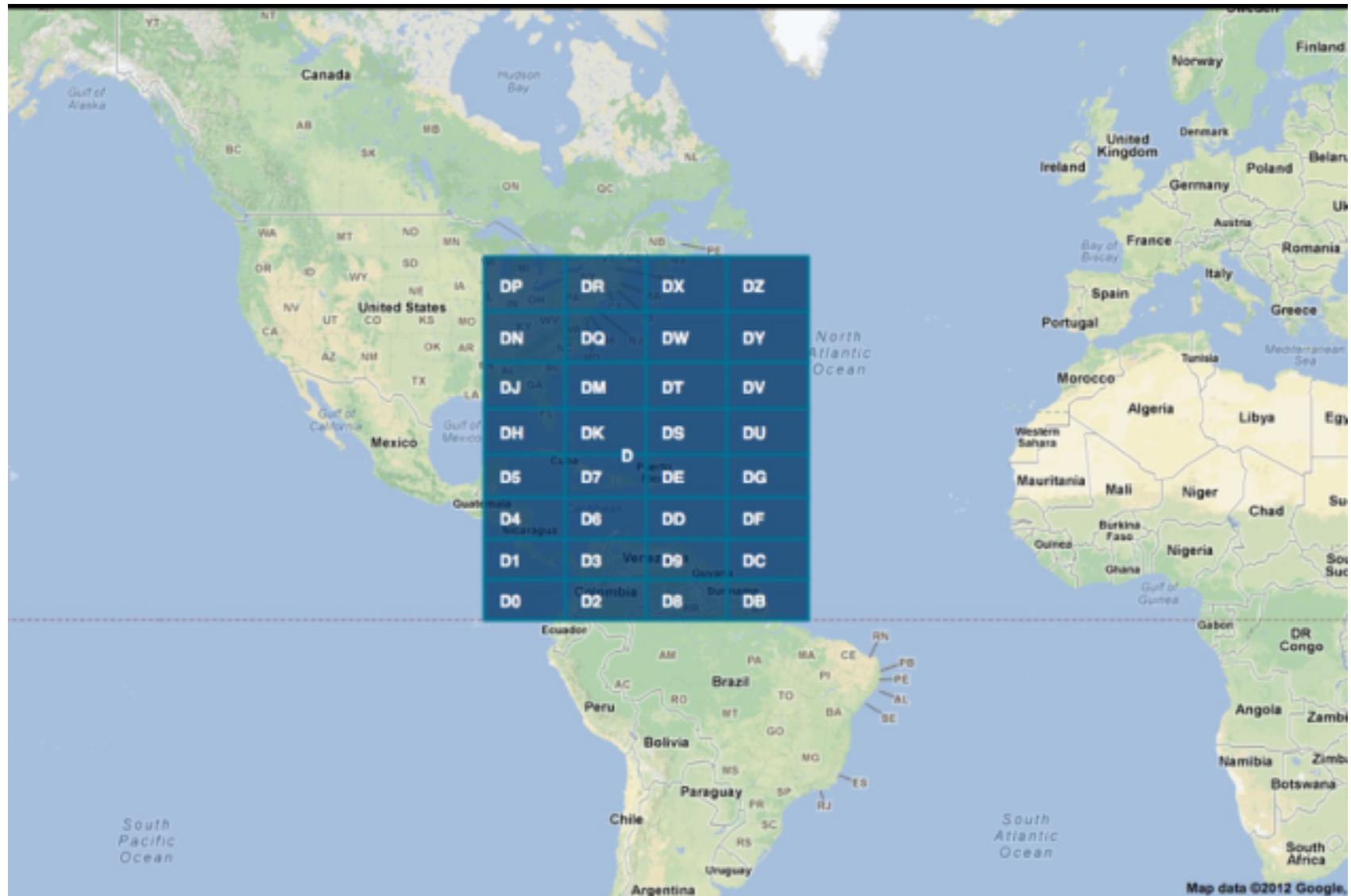
Hilbert Curve

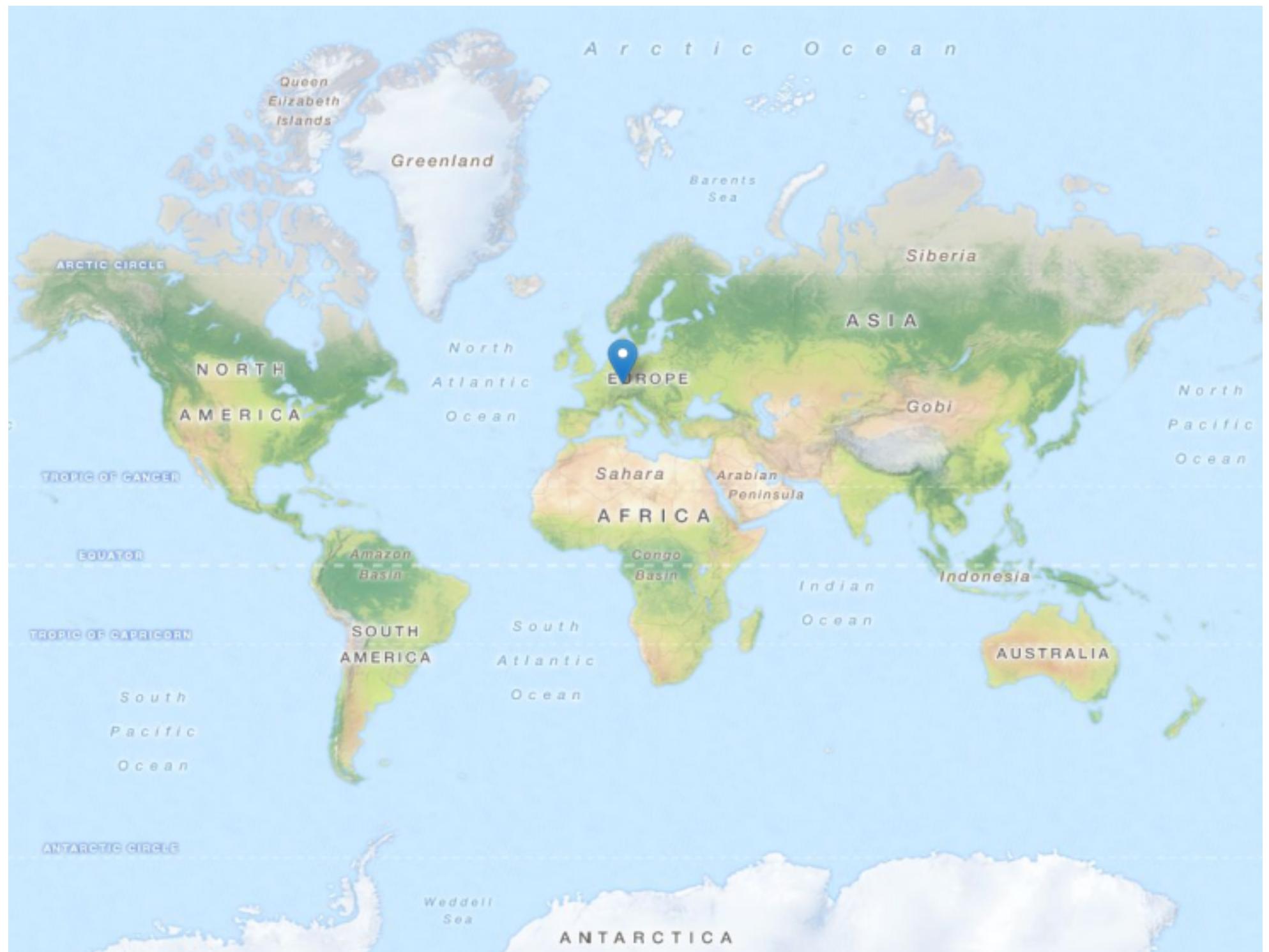


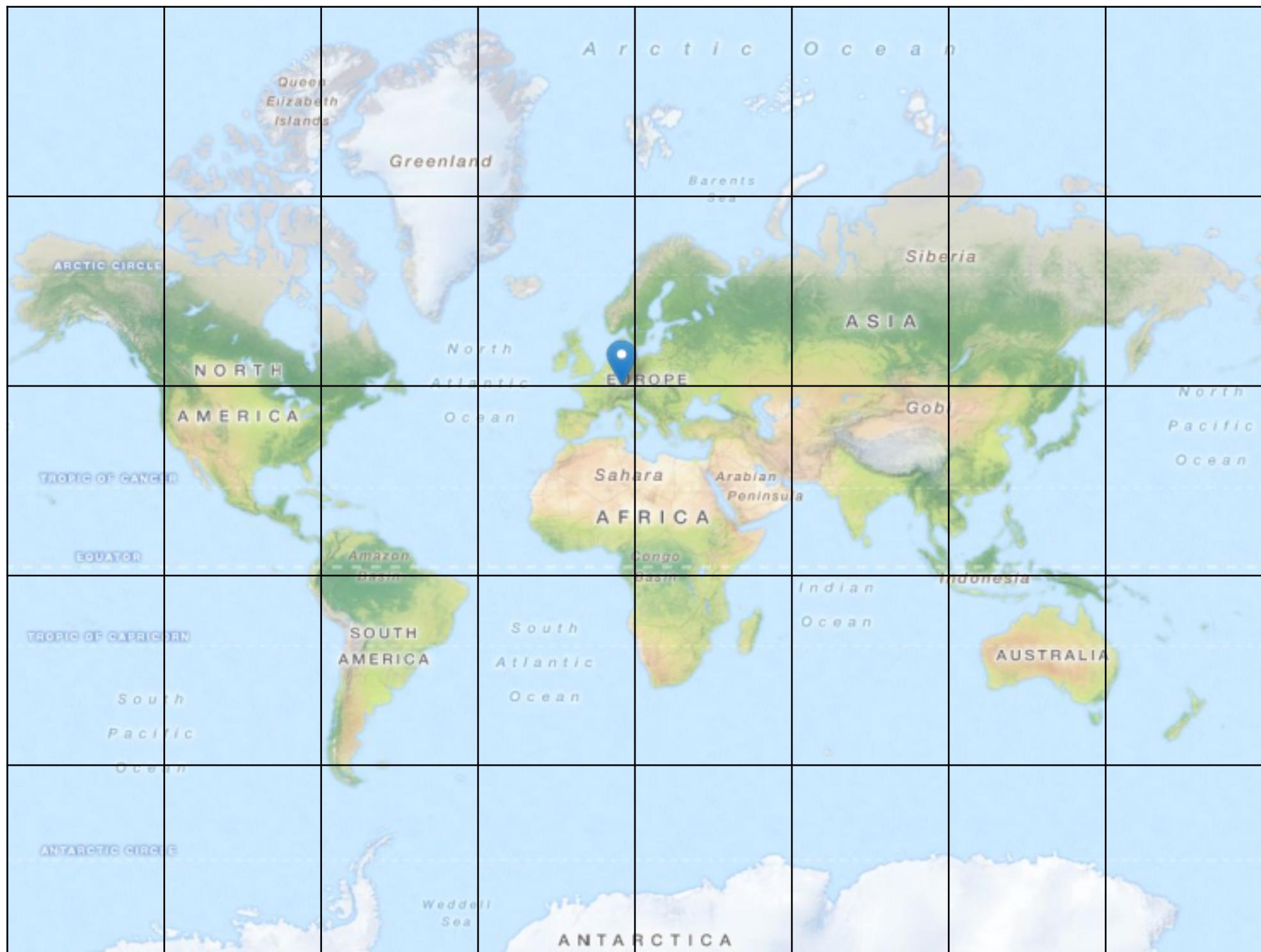
GeoHash

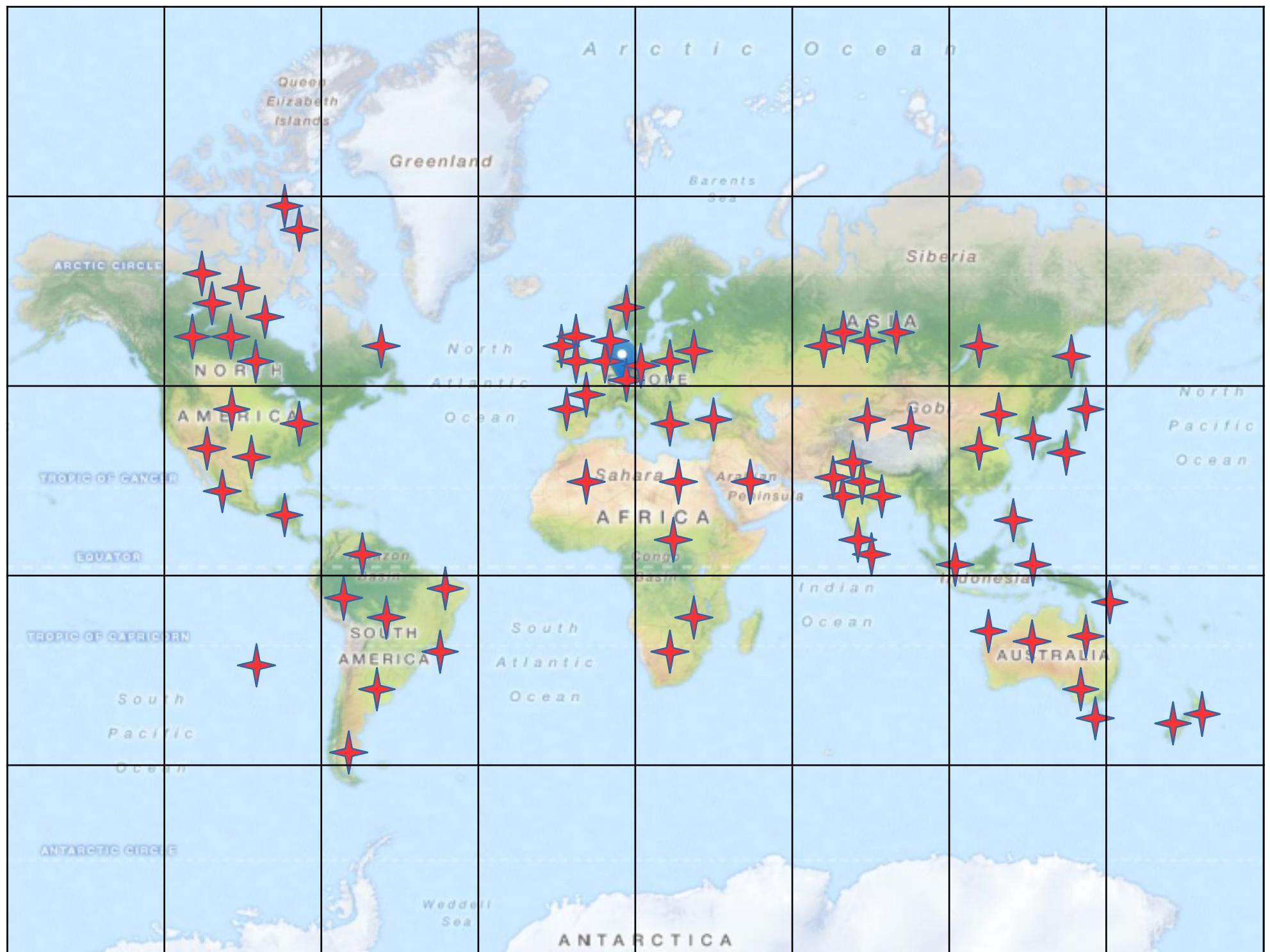


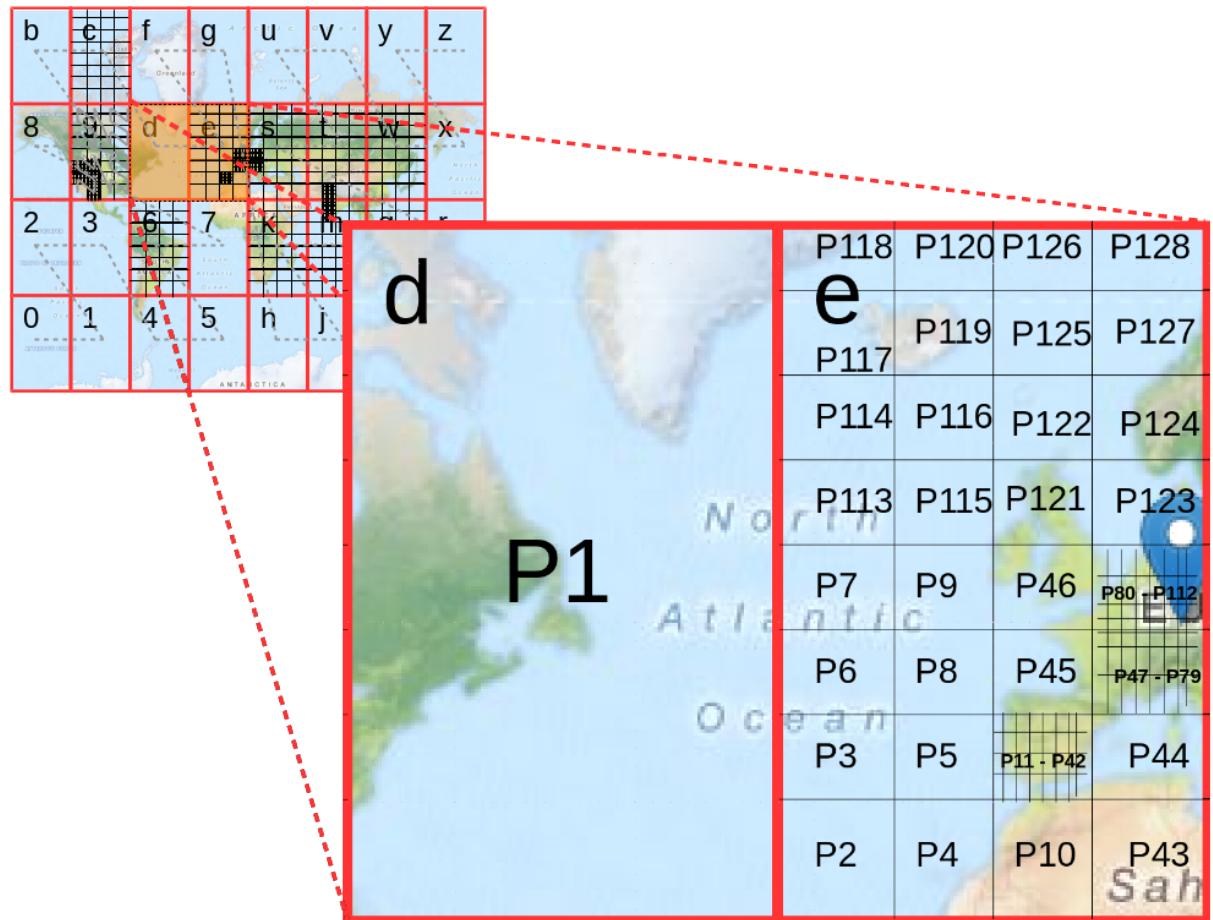
GeoHash





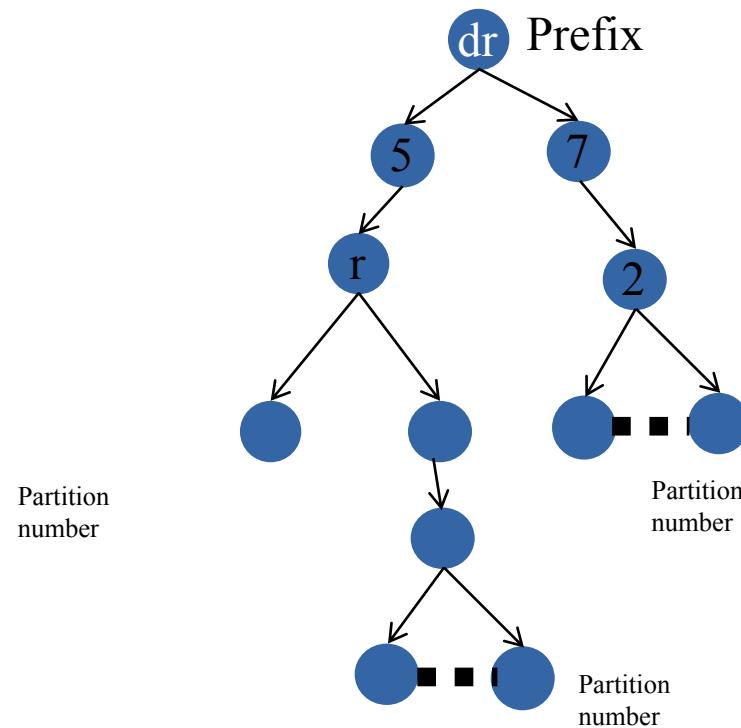


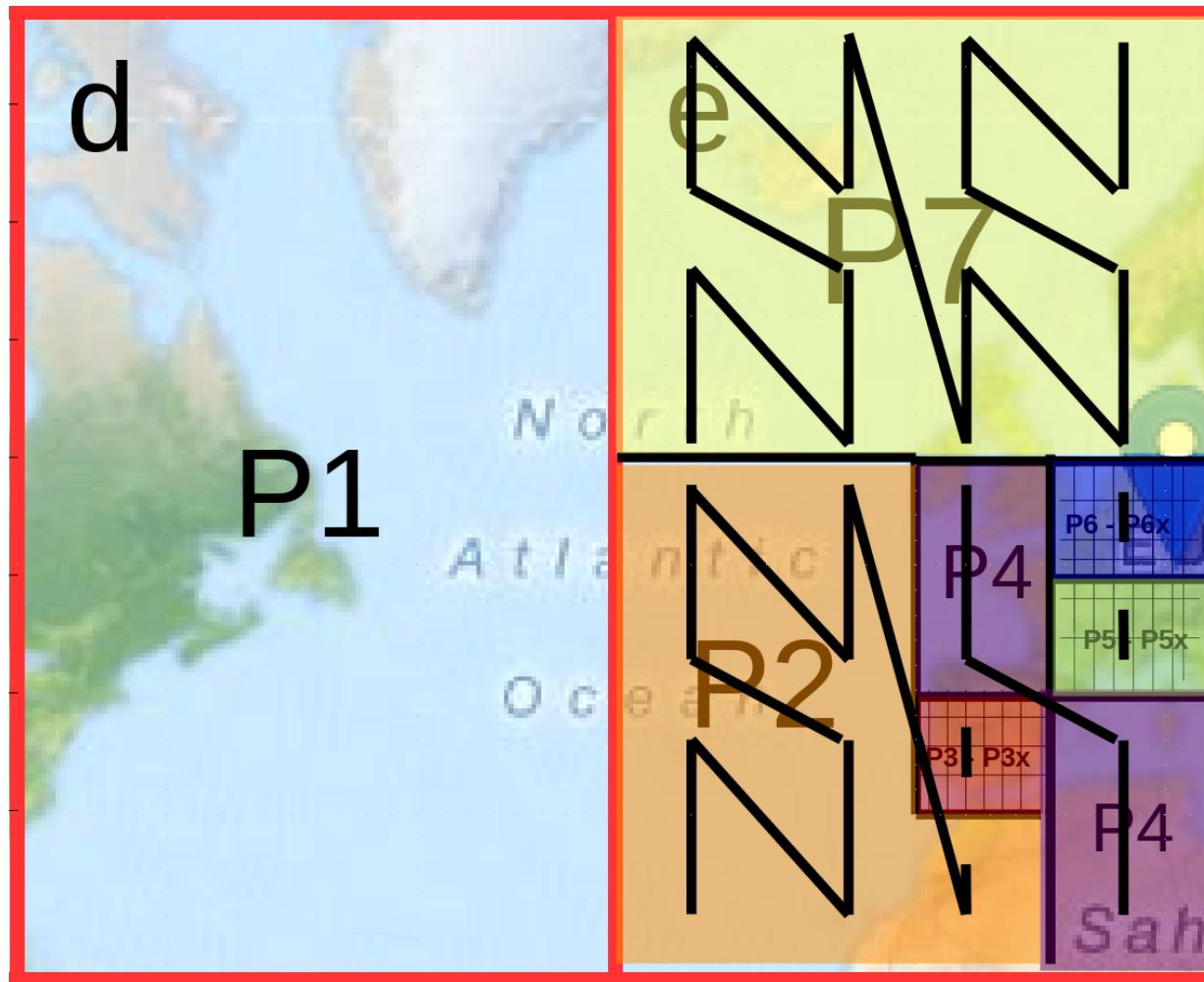




Variable length

- ❖ So now we need to map to nodes
- ❖ This variable length grids are mapped on to a data structure

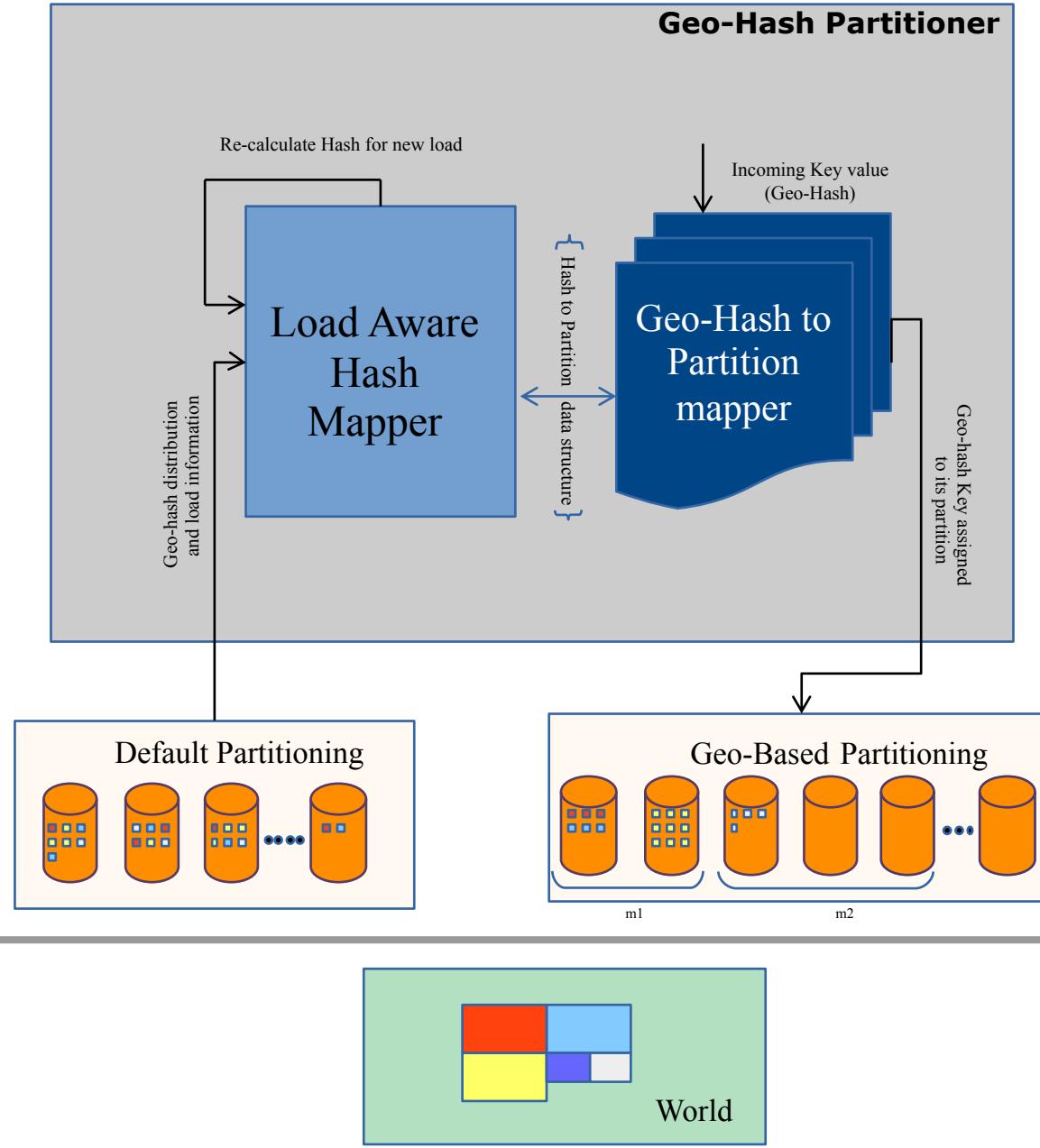




Custom Partitioning

- ❖ Step1: Load Aware
- ❖ Step2: Variable length to generate Hash
- ❖ Step3: Map these hash to RDD partitions
- ❖ Step4: Distribute or lay out data accordingly

```
import org.apache.spark.Partitioner ←  
  
class GeoHashPartitioner(partitionerMap: Map[String, Long])  
    extends Partitioner {  
    ...  
  
    def numPartitions: Int = partitionerMap.size  
  
    def getPartition(key: Any): Int = key match {  
        ...  
        partitionerMap.find(i => keyPref.startsWith(i._1))  
            .getOrElse(default)._2.toInt  
    }  
  
    override def equals ....  
  
    override def hashCode ....  
}
```



Region with less data points
Neighboring Region with less data
Region with no data points



Region with a lot of data points ($>$ threshold)
Smaller region with a lot of data points

Efficient Spatial processing with Spark

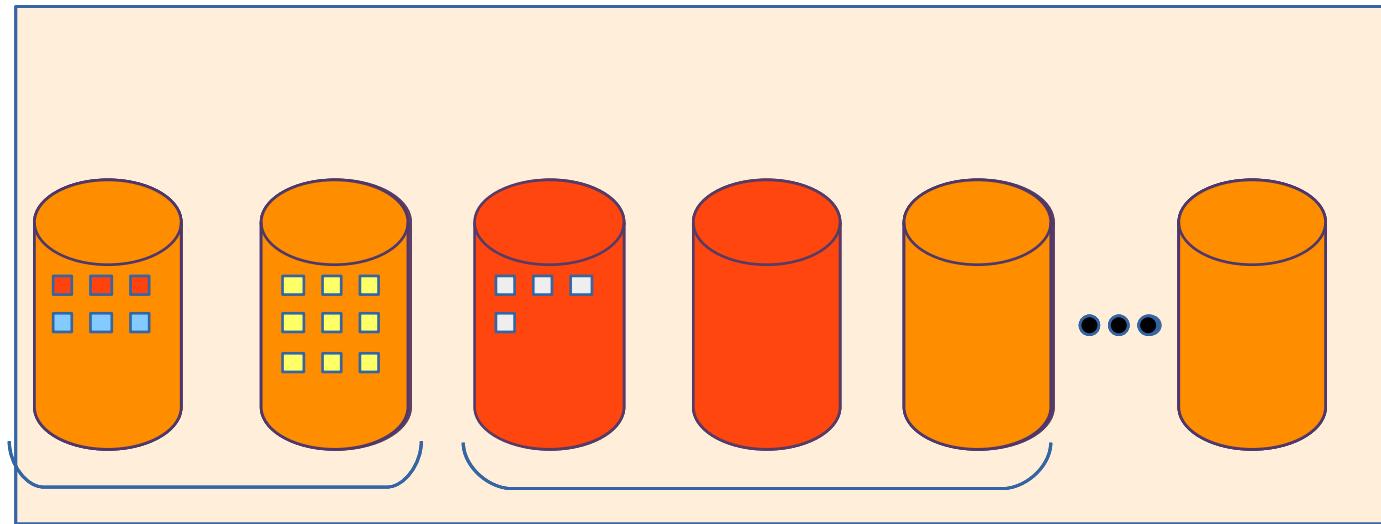


TECHNISCHE
UNIVERSITÄT
DARMSTADT

- ❖ GeoHash Partitioner
- ❖ GeoHash Query Layer

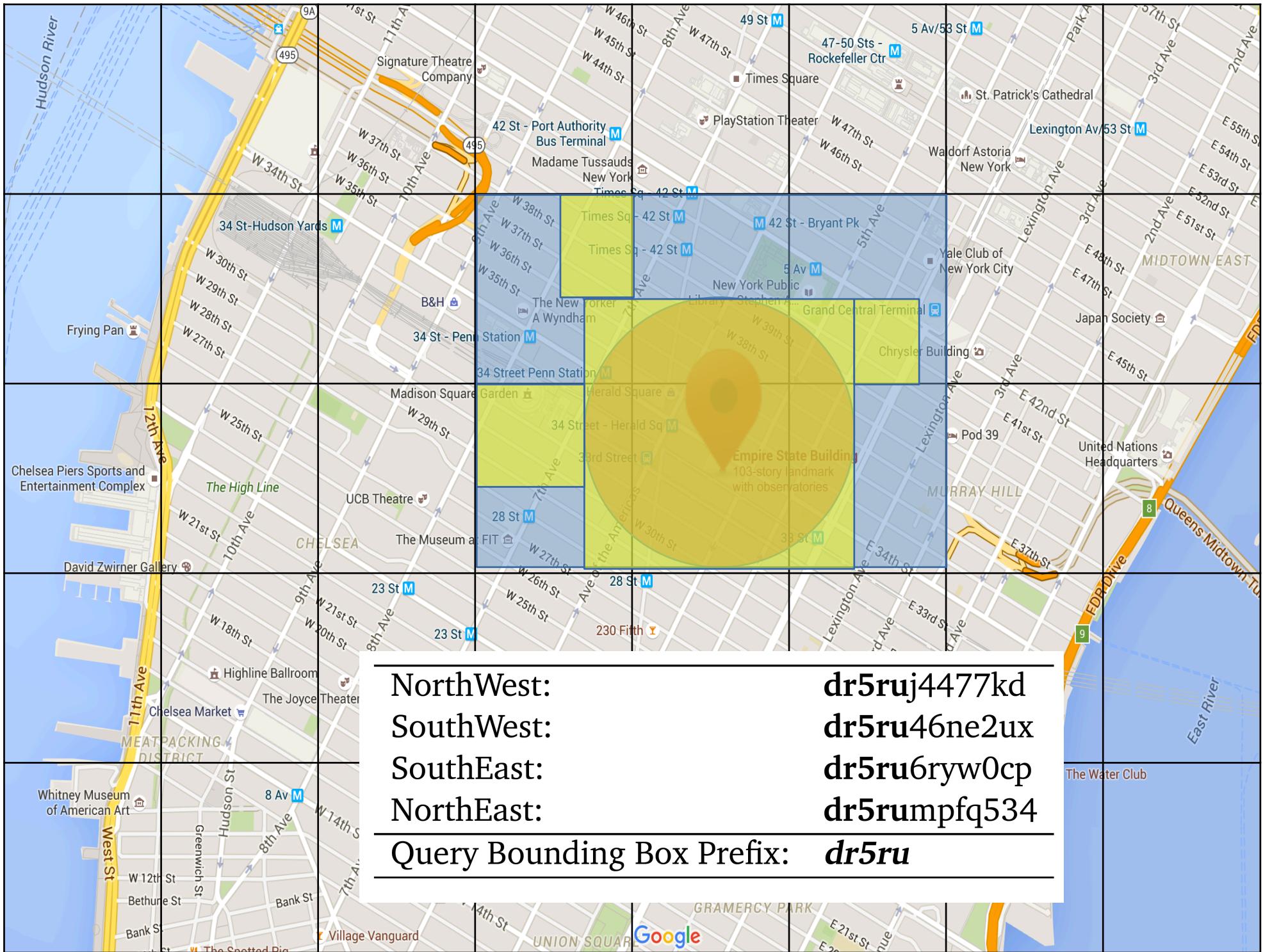
Query act on all nodes

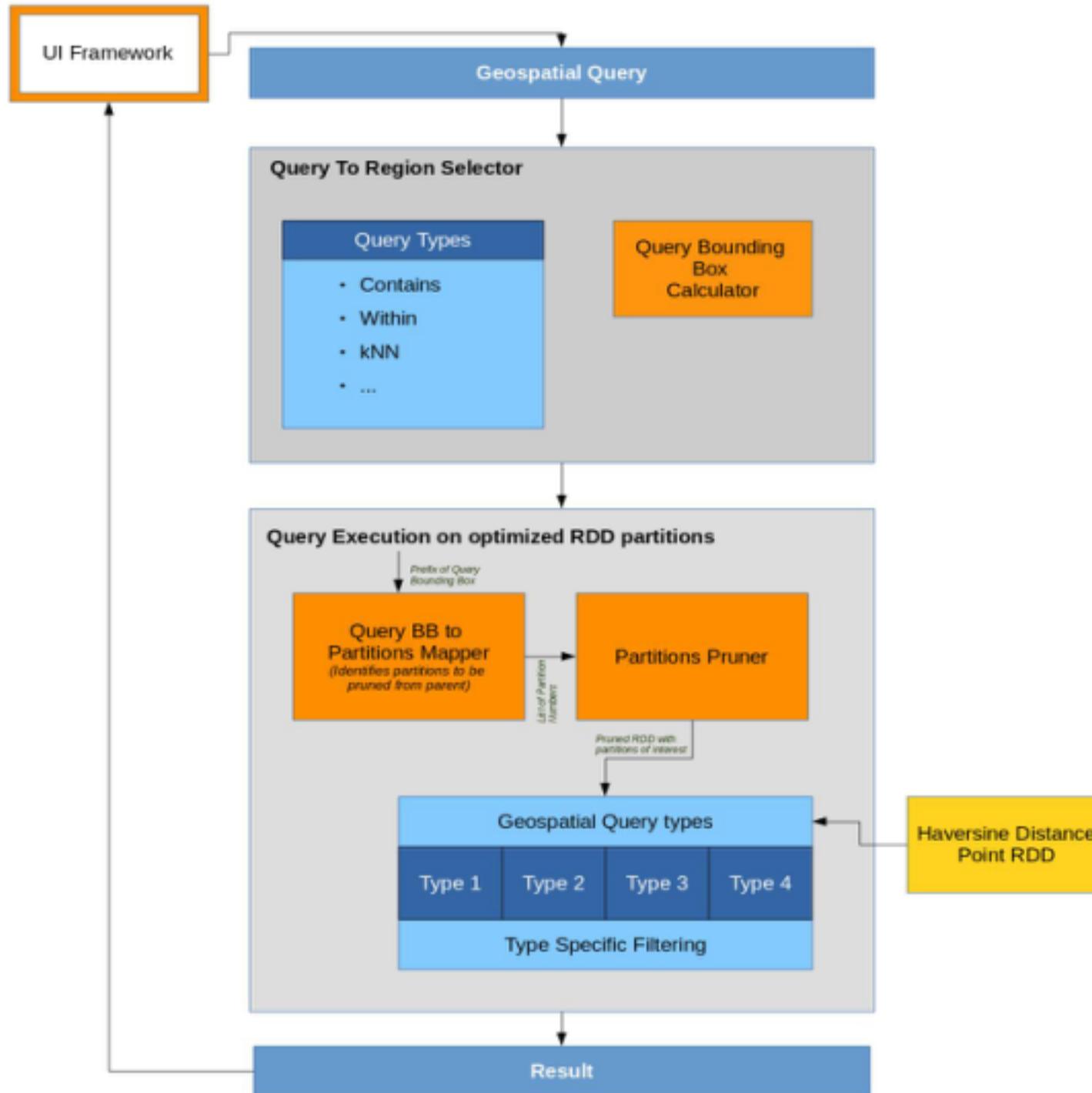
- ❖ Normally executes code/query on all partitions of nodes
- ❖ Large scans
- ❖ Resource utilization



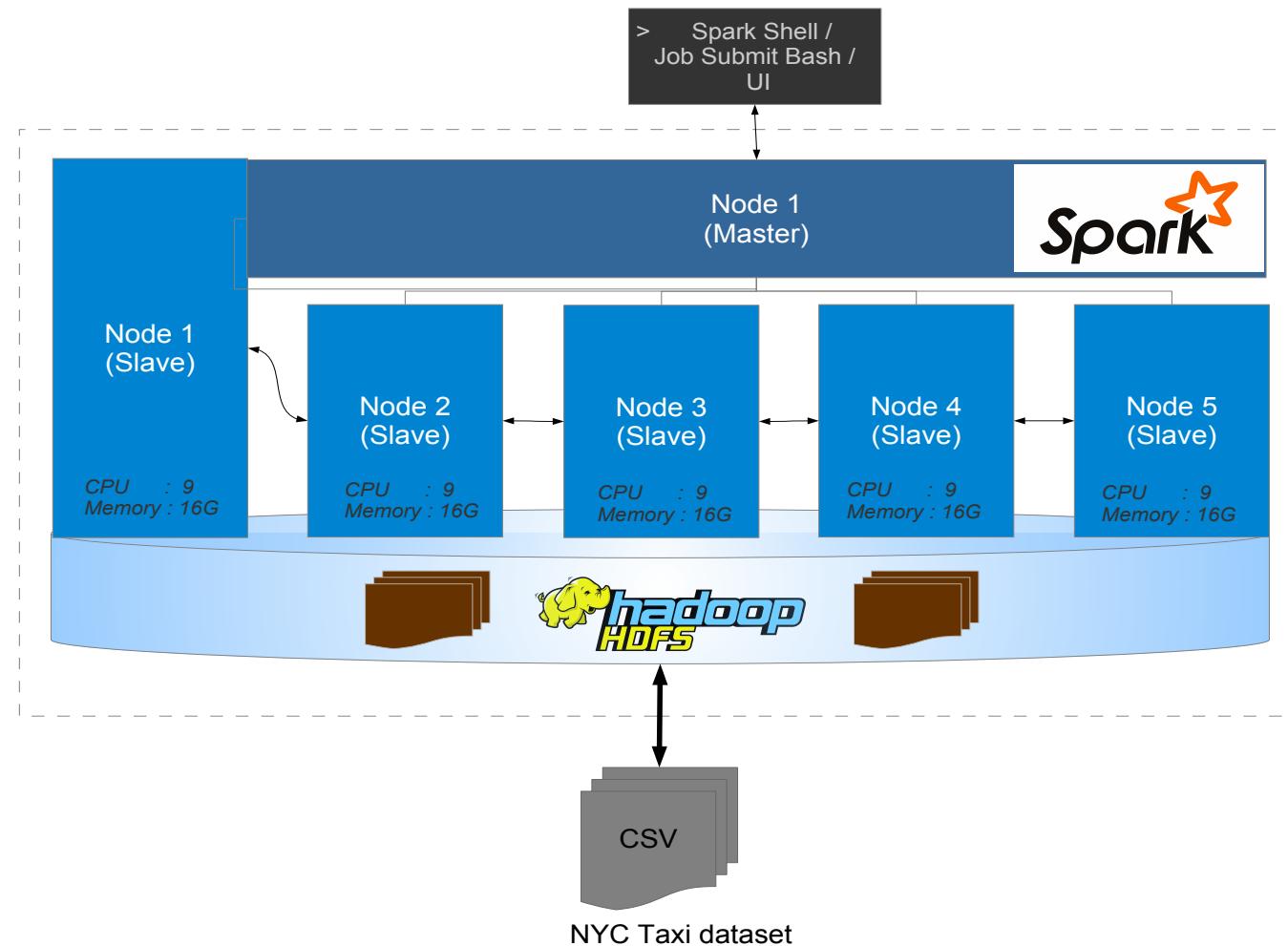
GeoHash Query Layer

- ❖ Leverage the **knowledge of previous partitioning** and efficiently query on subset of data:
- ❖ Identify the **type of Query**
- ❖ Determine the **areas of interest** from the overall world
- ❖ Figure out partitions that contain our interested region
- ❖ Slice out the partitions from the RDD
- ❖ Push the query on the final RDD based on the Query predicates

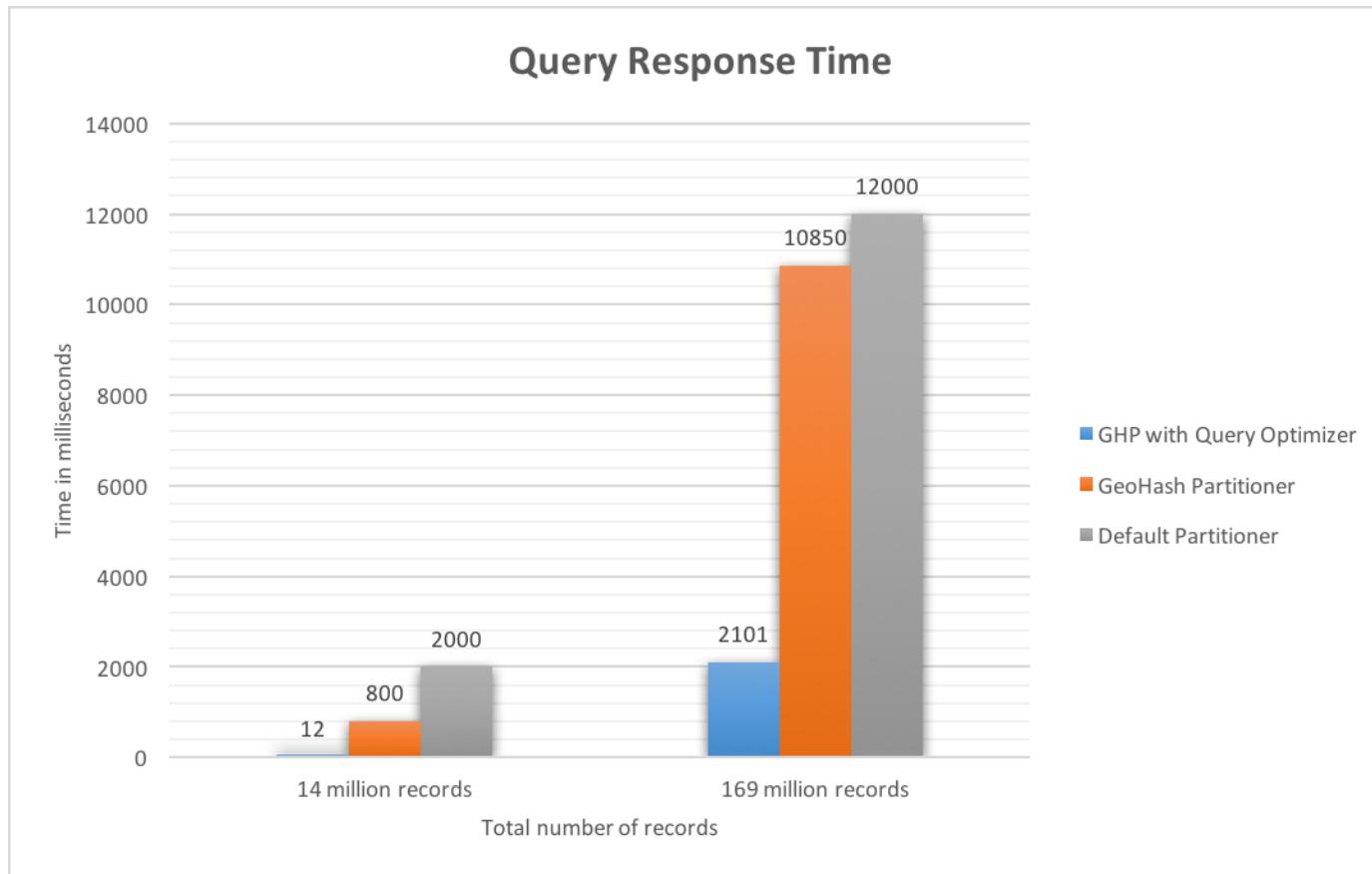




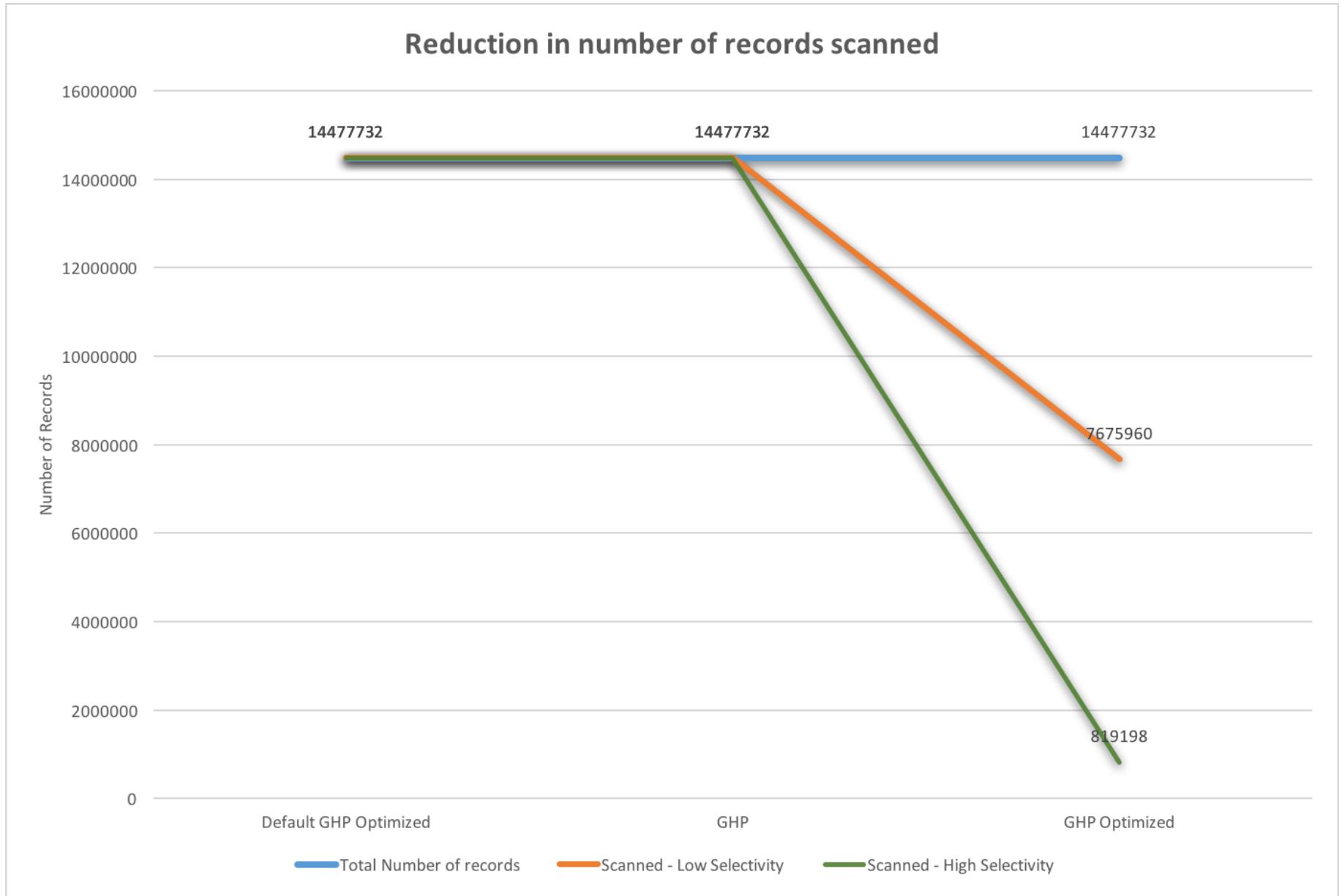
Evaluation Results



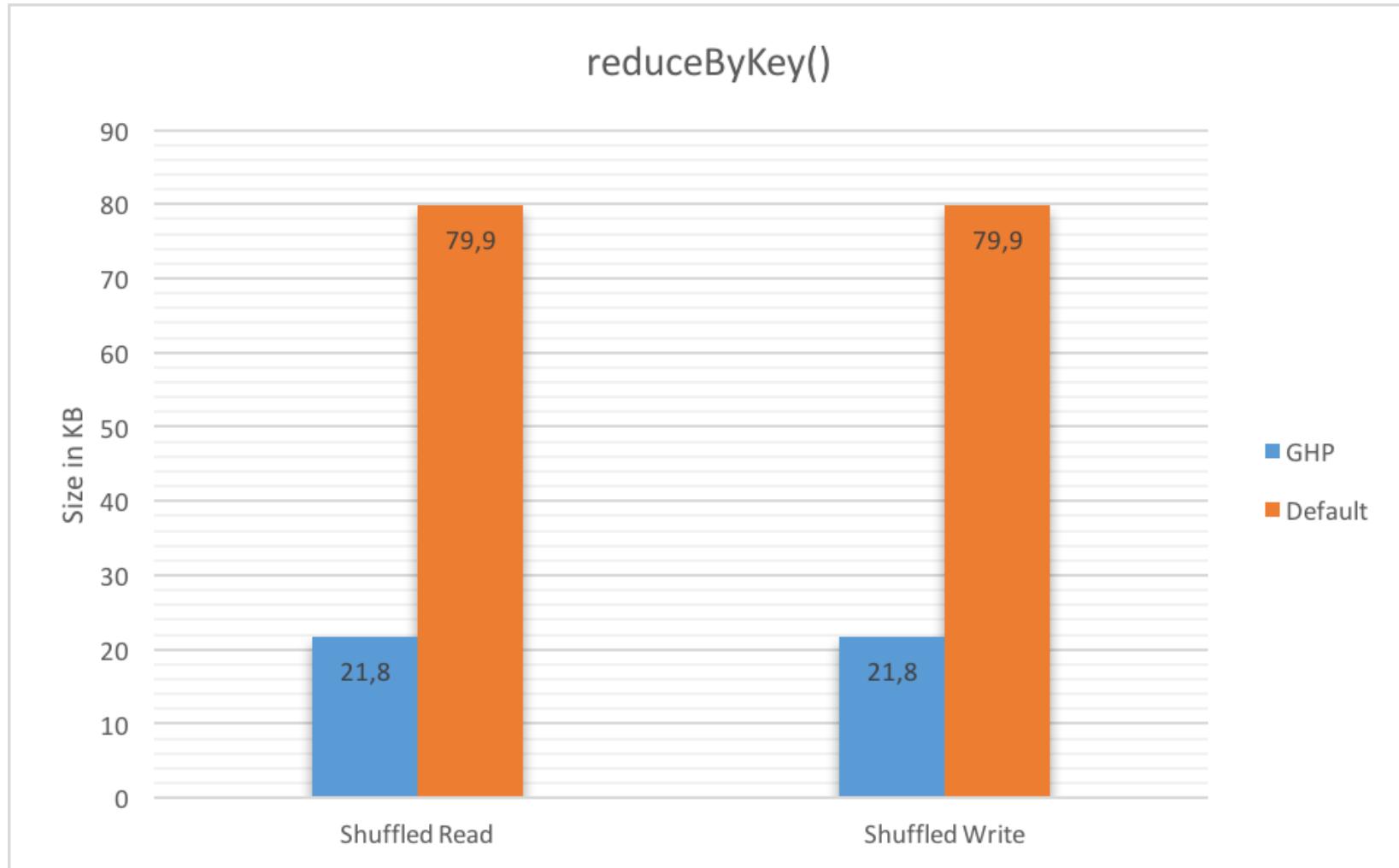
Improvement in Spatial Query Response time



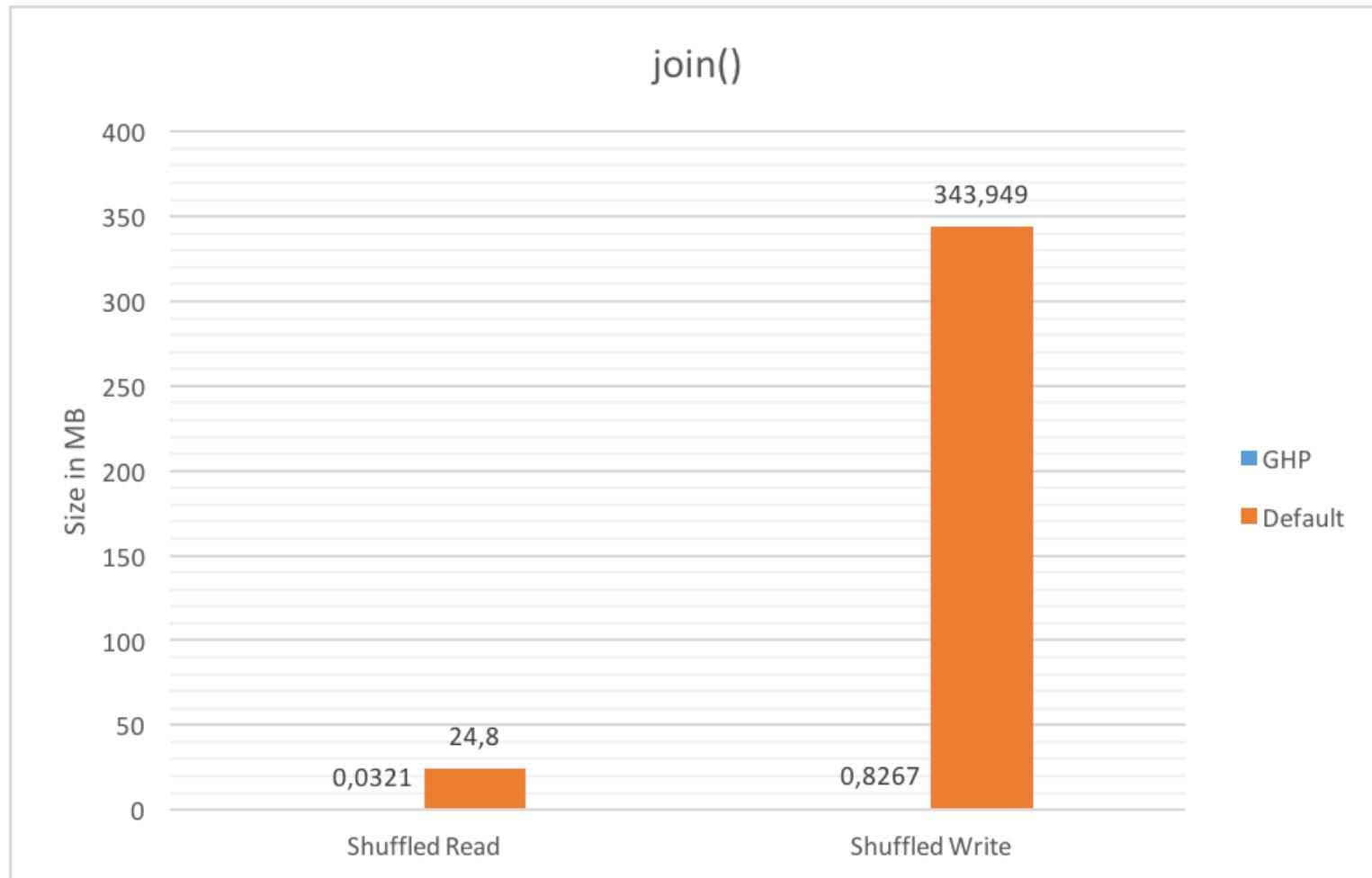
Reduction in number of records scanned



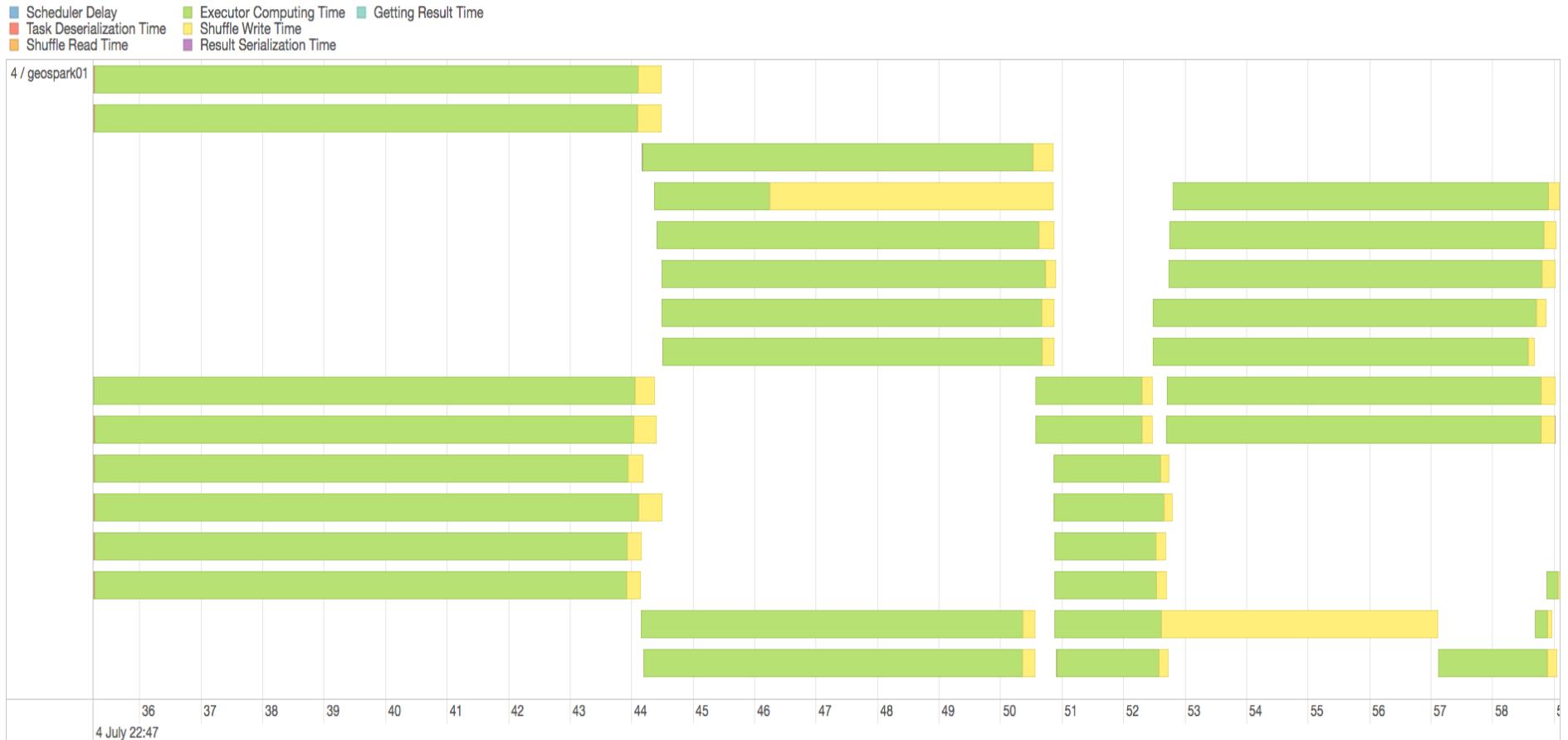
Reduction Locality of Spark tasks and minimized shuffling



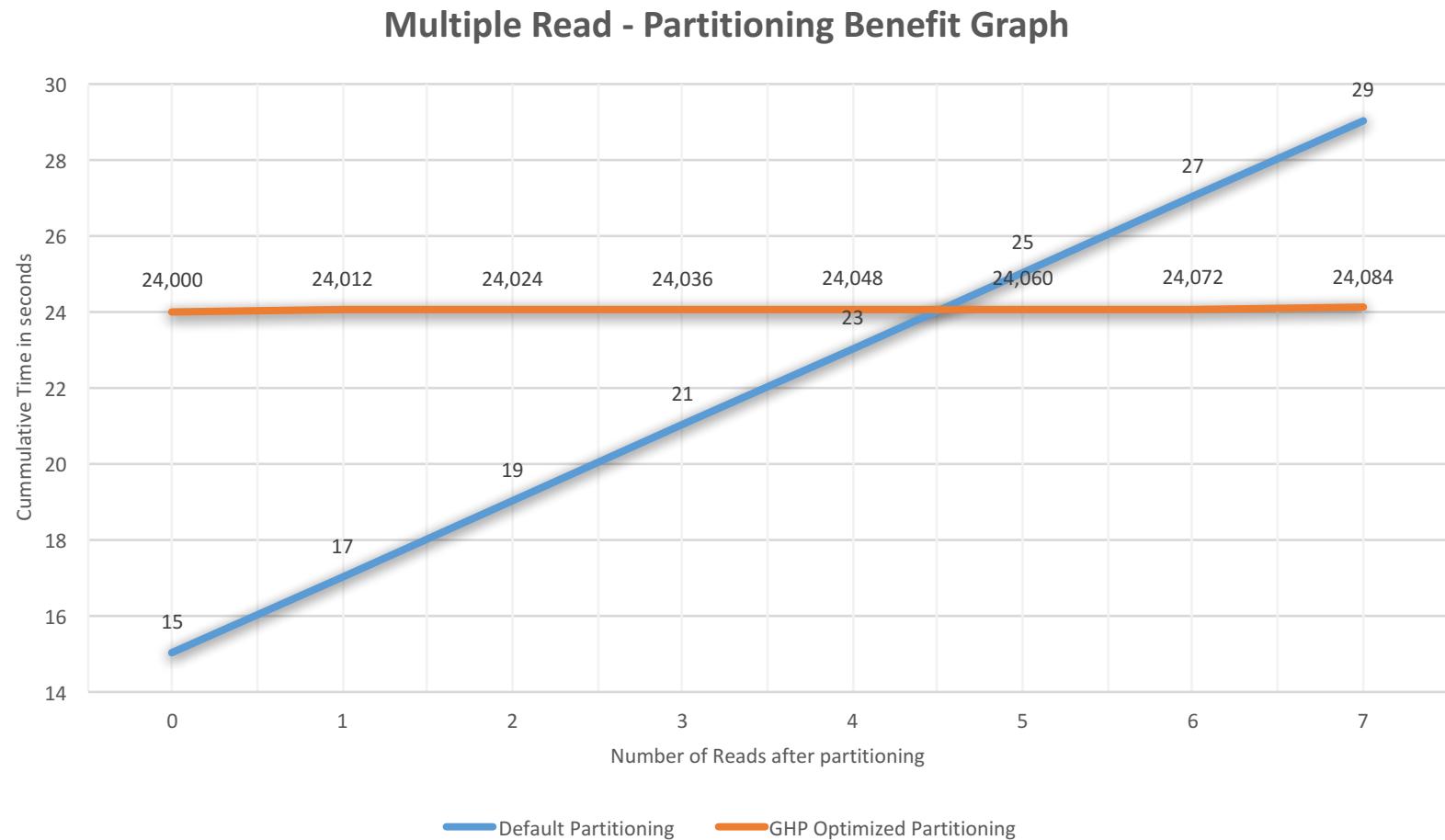
Reduction Locality of Spark tasks and minimized shuffling



Reduction Locality of Spark tasks and minimized shuffling



Partitioning is not always beneficial: Benefits of Partitioning with multiple Reads

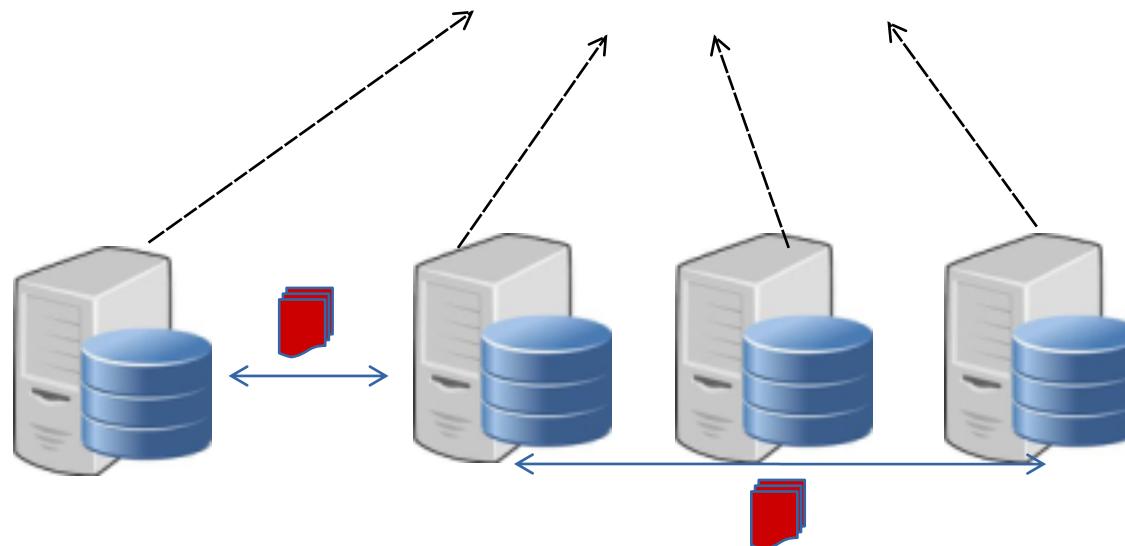


Problem is ~~clear~~ addressed

For a query like,

"Give me all tweets activities that happened around my campus (say within 300 meter)"

- ❖ Tweets with geo tag TUDarmstadt is ~~distributed across all partitions~~ in the cluster
- ❖ Shuffles them between partitions (and in turn between nodes)
- ❖ ~~Scans a larger section (say, records of whole Germany)~~



Project Details



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Project page visit: <http://hariharan-gandhi.github.io/thesis>

References



1. Facebook logo - <https://www.facebookbrand.com/>
2. Snapchat logo - <https://www.snapchat.com/brand-guidelines>
3. Twitter logo - <https://about.twitter.com/company/brand-assets>
4. Instagram logo - <https://www.instagram-brand.com/>
5. WhatsApp logo - <https://www.whatsappbrand.com/>
6. IBM logo - <https://www-03.ibm.com/press/us/en/photo/20190.wss>
7. Geotagging - <http://smbp.uwaterloo.ca/wp-content/uploads/2014/07/geotagging.jpg>
8. Map tile screenshot - ©2016 MapQuest <http://open.mapquest.com/> | ©OpenStreetMap contributors - <http://www.openstreetmap.org/copyright> | ©Google Maps <maps.google.com>
9. Statistics: Location based service - <https://wangxiaoyujade.wordpress.com/2015/02/25/location-based-service-groupon/> Source: Socintel360 as cited in company blog <http://www.emarketer.com/>
10. Statistics: mobile users - GSMA, "Gsma: The mobile economy 2016." <http://www.gsmamobileeconomy.com/>
11. RDD partitionby effect - <https://www.safaribooksonline.com/library/view/learning-spark/9781449359034/ch04.html#idp10177456>

Thank you



TECHNISCHE
UNIVERSITÄT
DARMSTADT



- Prof. Alejandro Buchmann Ph.D.
 - Robert Rehner, M.Sc.
 - Dr. Gregor Möhler (IBM, Germany)
 - Dr. Raghu Kiran Ganti (IBM Lab, USA)
-



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Master Thesis Student:

- Hariharan Gandhi