Geospatial Analytics at Scale

Milos Milovanovic
Co-Founder & Data Engineer
milos@thingsolver.com

THINGS SOLVER | Things Solver
ENLIGHTEN YOUR DATA

What we do?

Advanced Analytics company based in Belgrade, Serbia.

We are operating worldwide in effort to find the most valuable approaches and solutions for handling data.

Things Solver projects across following industries:

- Telecommunication
- Banking
- Retail

Our clients:

vip Banca Intesa PLANETA SPORT

A1 Content Insights

Importance of Geospatial Analytics

Spatial context of the data is an important variable in many advanced analytics applications.

Massive volumes of spatial data generated on daily basis.

- Sensor data
- GPS
- Transportation and movement
- Weather data
- IP addresses
- Search data
- Social media
- ...

Spatial context provides important insights into behavior and mobility which drives a smarter approach to decision making.

Applied Geospatial Analytics

Industry

- Telco, Travel Services, Search Engines, Applications, Advertising

Public Safety

- Law Enforcement, Civil Security and Public Safety Organizations

Disaster Management

- Urban disaster risk, Crisis response, Advocacy and Human Rights

Climate Change Adoption

- Climate change impacts and trends, environmental modeling

Social Sciences

- Spatial dynamics, economics, modelling human-environmental interaction

tripadvisor®

UBER

NASA

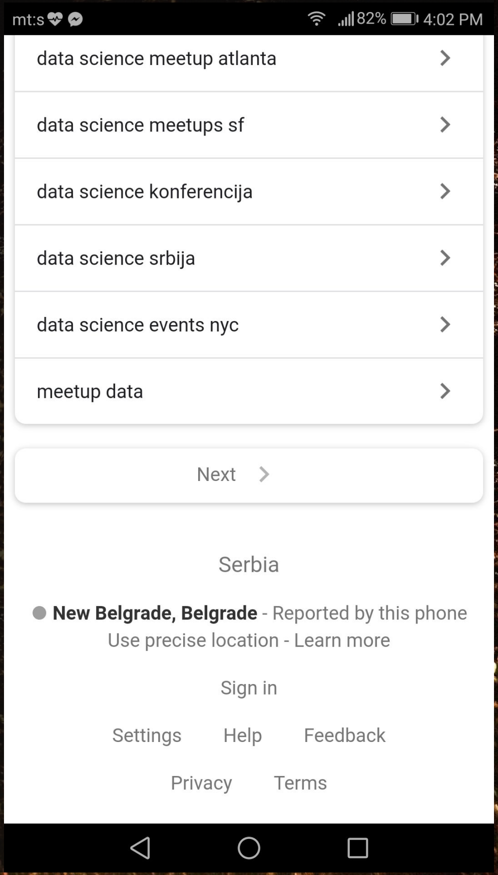
WWF

European Environment Agency

Ushahidi

Questions that Spatial Analysis can answer

- Show me all the clubs in this neighbourhood?
- How many people pass by this billboard per day?
- What is the commute trend on daily basis per some city?
- Which adds should I place for people living in particular area?
- In which areas my mobile subscribers have the network problems?
- How much time will I need to get to location A?



data science meetup atlanta

data science meetups sf

data science konferencija

data science srbija

data science events nyc

meetup data

Serbia

● **New Belgrade, Belgrade** - Reported by this phone
Use precise location - Learn more

Geospatial Modeling Challenges

Complex data types

Complex GEO formats

Operations

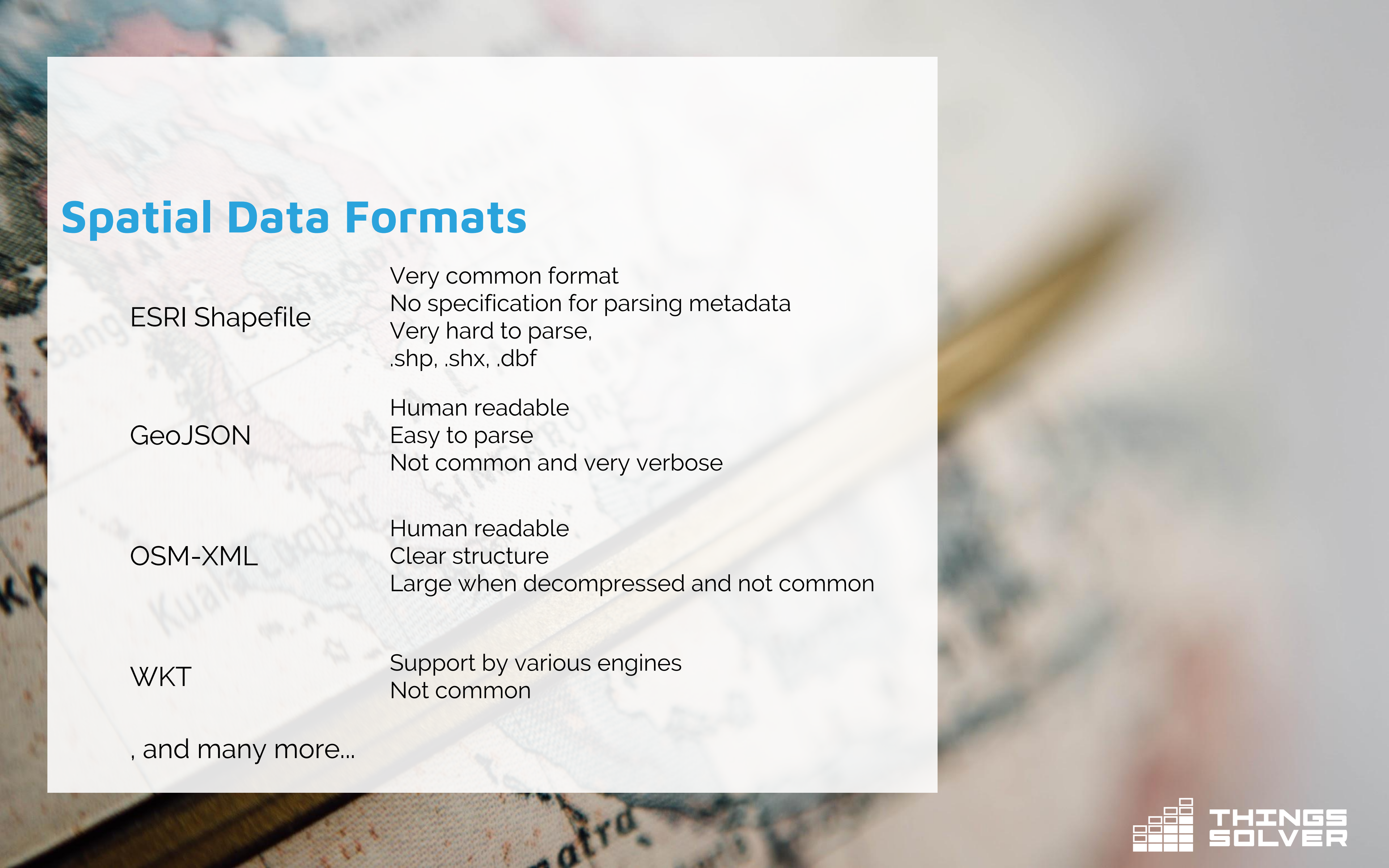
Various coordinate systems

Accuracy

Vast amount of data

Spatial Data Types

Point	Location Event
Line	Path Trajectory Road
Polygon	City Region Area
Curve	Continuous path/trajectory



Spatial Data Formats

ESRI Shapefile	Very common format No specification for parsing metadata Very hard to parse, .shp, .shx, .dbf
GeoJSON	Human readable Easy to parse Not common and very verbose
OSM-XML	Human readable Clear structure Large when decompressed and not common
WKT	Support by various engines Not common

, and many more...

Spatial Operations

Contains

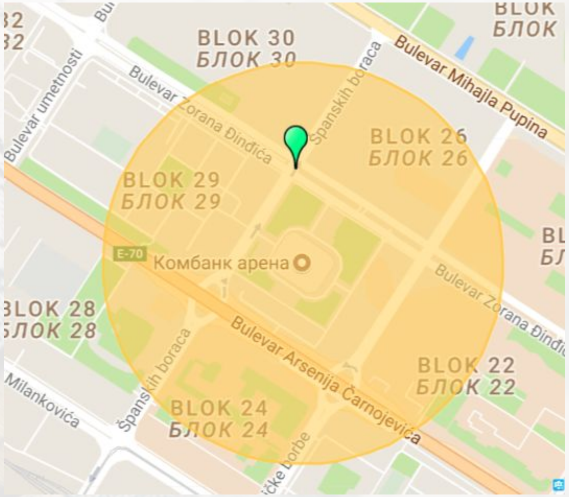
Within

Intersects

Intersection

Spatial joins

Indexing

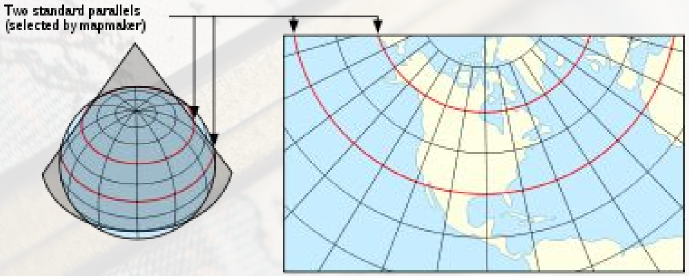


Coordinate Systems

Cartographic projection systems usually do not contain GPS coordinates - they are trying to be more precise.

Converting latitude/longitude information to various coordinate systems and vice-versa is not trivial.

Some examples: WGS84, UTM, UPS, MGRS, ...



Lambert Conformal Conic Projection

Open Source Tools and Challenges

Parse vast amount of spatial data at scale. (Shapely)

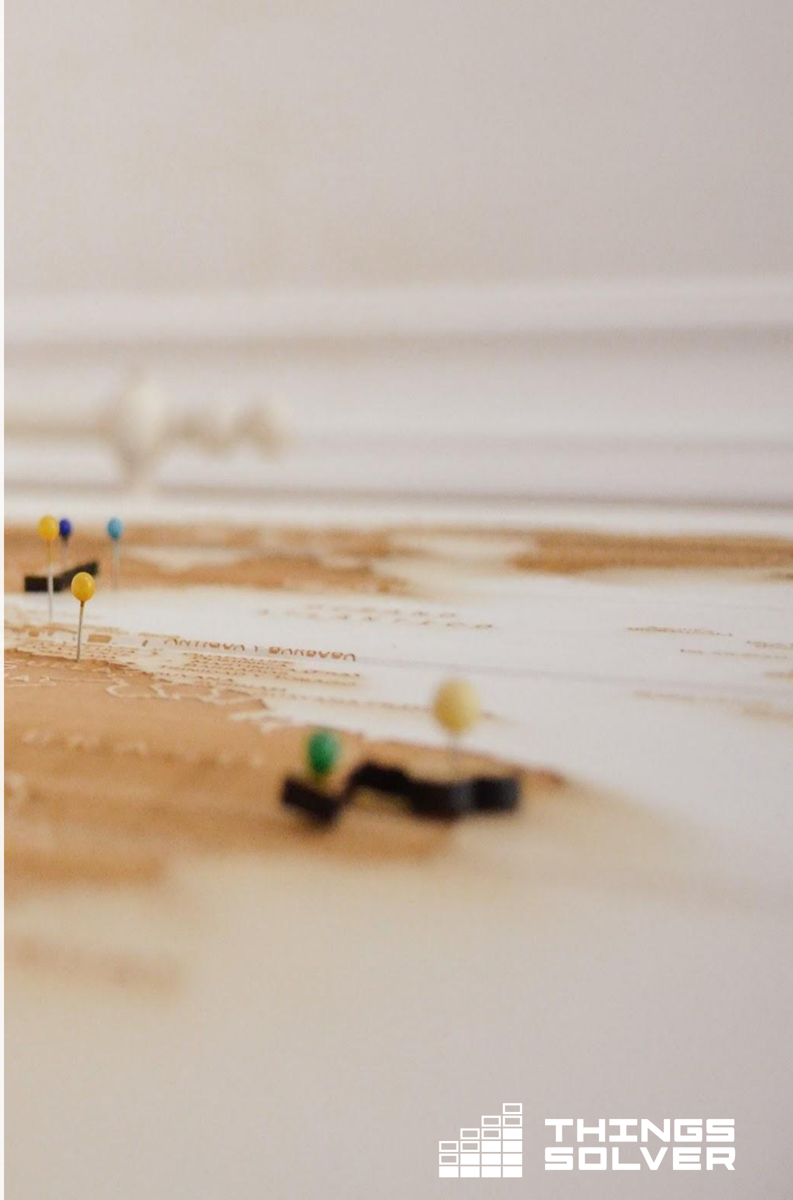
Spatial joins. (ESRI Hive)

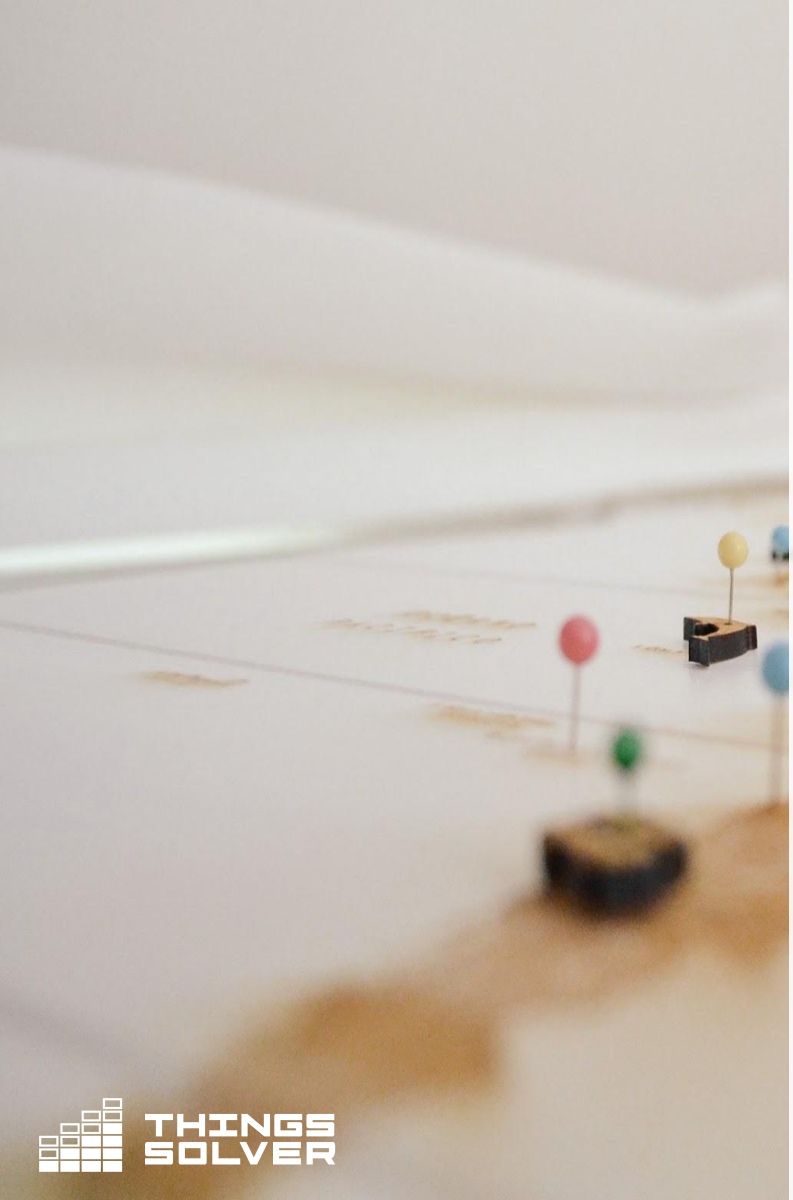
Complex APIs. (GeoSpark)

Read various data formats.

Work efficiently with geo data.

Enable exploration.





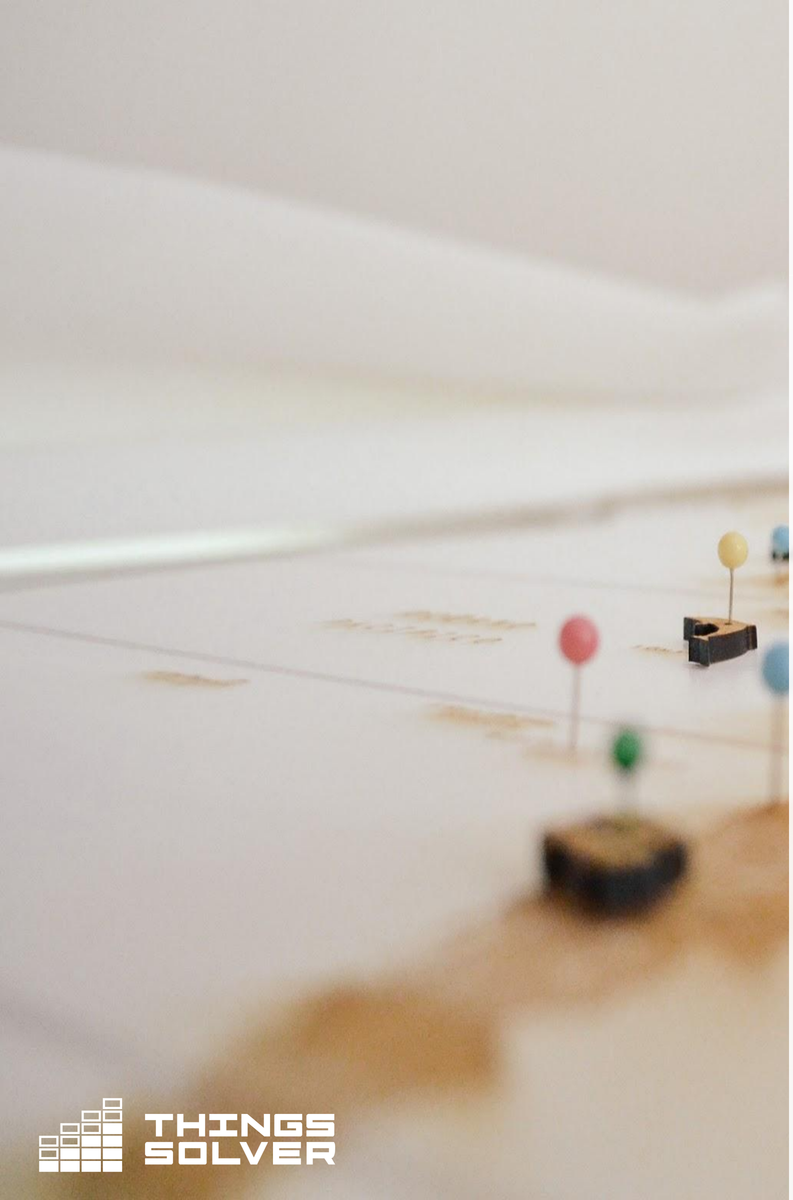
Geospatial Analytics on Spark

Apache Spark is a very fast processing engine utilized for efficient analysis of vast amount of data.

Provides rich APIs in Scala, Python and R, and is applied in many applications and architectures.

SQL interface through DataFrames.

Catalyst optimizer provides high performance.



Geospatial Analytics on Spark

There are a few libraries built on top of Spark for working with the spatial data:

- **GeoMesa.** A suite of tools for persisting, querying and analyzing spatial data at scale. It provides optimized spatial SQL for data manipulation. GeoMesa provides very rich API for geospatial analysis. Cons: massive and slightly massive configuration.
- **GeoSpark.** GeoSpark extends Apache Spark with a set of out-of-the-box Spatial Resilient Distributed Datasets (SRDDs) that efficiently load, process, and analyze large-scale spatial data across machines. The project is growing rapidly to support various partitioning, indexing, distance and neighbor functions. It provides GeoSparkViz to visualize Spatial RDD and Spatial Queries.
- **Magellan.**

THINGS SOLVER

Magellan

Distributed execution engine for geospatial analytics implemented on top of Apache Spark to scale out computation.

It deeply leverages modern database techniques like efficient data layout, code generation and query optimization in order to optimize geospatial queries.

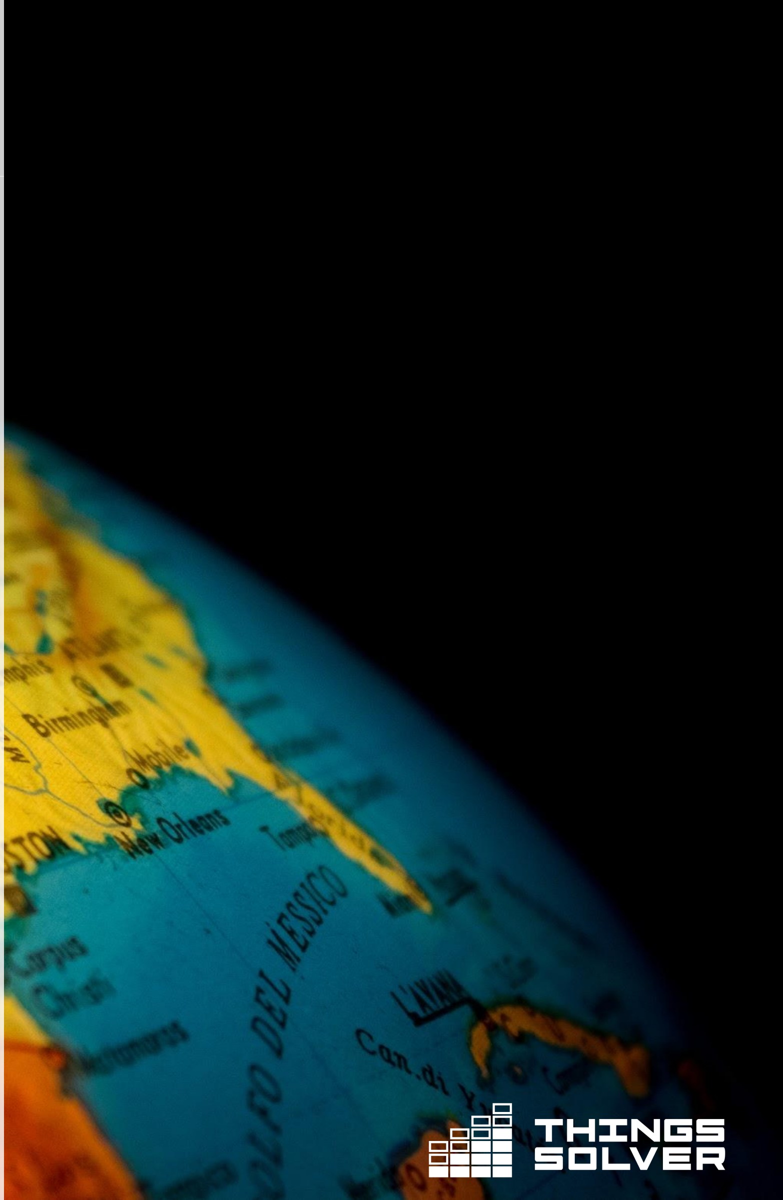
Magellan in general is set out to bring:

- Simplicity and intuitiveness
- Scalability
- Flexibility and extendability

with Catalyst optimization and Data Frames. It allows:

- Writing applications in your favourite language (Scala, Python, R)
- Simple API - writing less code in application
- High performance algorithms
- Query optimization handled by the Spark(Catalyst)

https://github.com/harsha2010/magellan



THINGS SOLVER

Magellan

Geospatial Types

Point
Line
Polyline
Polygon

```
val points = sc.parallelize(Seq((-1.0,
-1.0), (-1.0, 1.0), (1.0, -1.0))).toDF("x",
"y").select(point($"x",
$"y").as("point"))
```

Geospatial Formats

Shapefile (including metadata)
GeoJson
OSM-XML?

```
val df = spark.read.
  format("magellan").
  option("type", "geojson").
  load(path)
```

Geospatial Operations

Boolean Expressions:

- Contains
- Intersects
- Within

Binary Expressions

- Intersection

```
data.filter(Point(54.716865,25.28547
9) within $"polygon").show()
```

Joins and Indexing

(Broadcast) Cartesian Join
Geohash Join
Z-order curve indexing

```
points.join(polygons index
30).where($"point" within
$"polygon")
```

Magellan

Working with Magellan types feels like working with native Spark structures - Magellan provides SQL extension that defines literals and expressions to query the data in Spark SQL manner.

```
area.filter(
        Point(44.8153831,20.434975)  // literal
        within //expression
        $"polygon"
).show()
```

Magellan

ESRI Java API to avoid serialization overhead (Scala to SQL DataFrame serialization) and index rebuild for high performance.

Write declarative commands and Catalyst will deal with optimization.

Catalyst optimization for Magellan

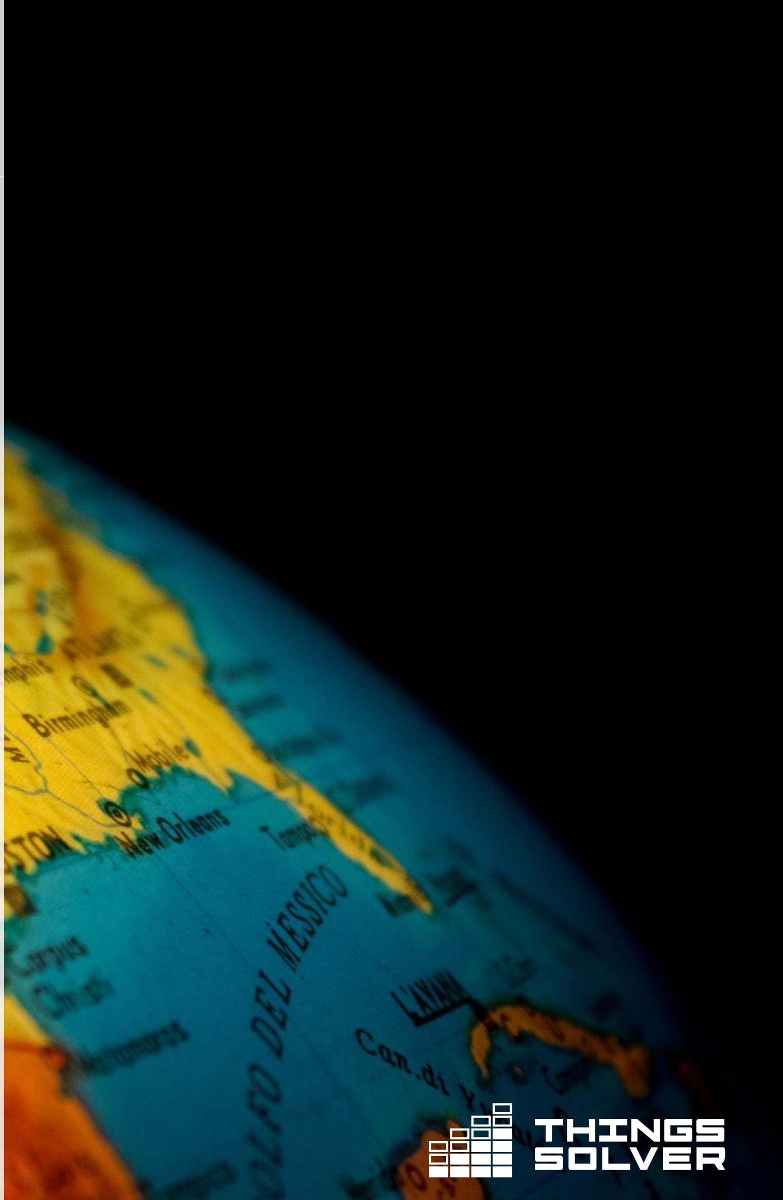
Join optimization - broadcast data to every node

If one of the tables is small => Broadcast Cartesian Join

Else => Cartesian Join

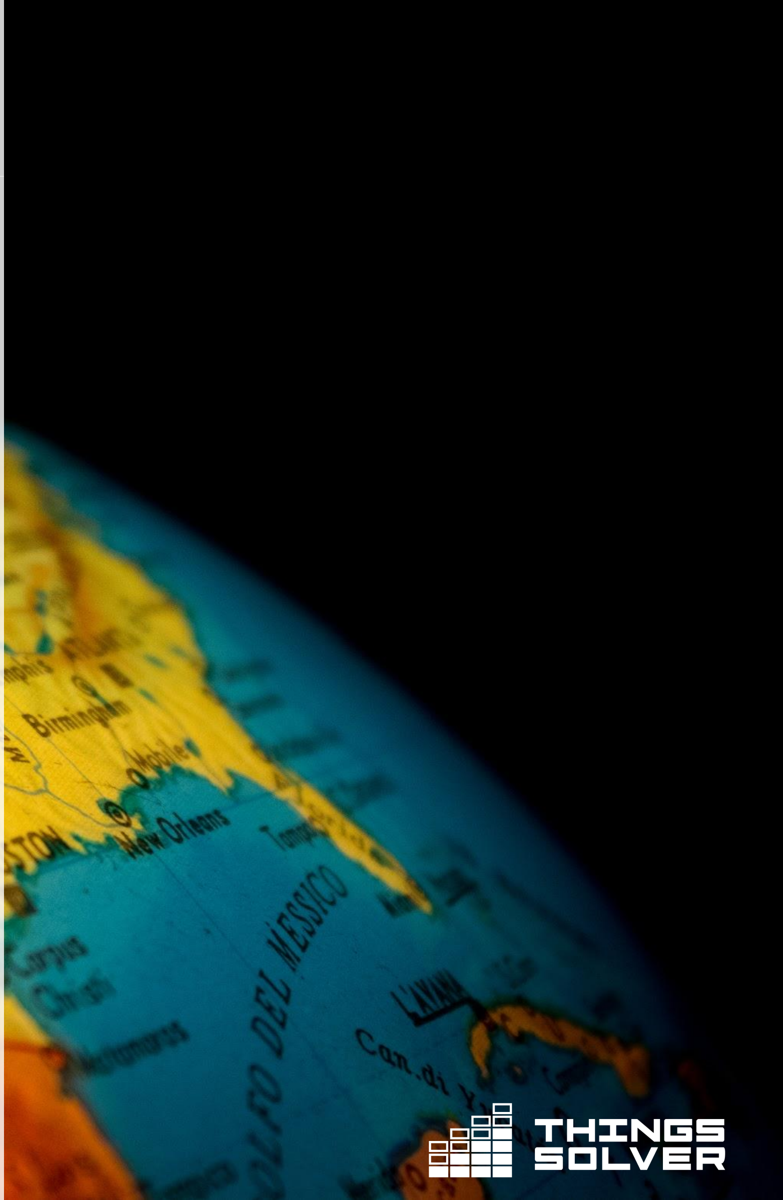
Z-order curve indexing + Geohash Joins

Override Spark SQL Join Strategy



Magellan - what is missing/expected

- Full Python support
- Distance operations
- Neighbor calculations
- Coordinate conversion (not to be implemented in general due to Apache SIS)
- Support for other data formats
- Performance and operations improvement
- Export to spatial format



DEMO

Geospatial Analytics at Scale

Milos Milovanovic
Co-Founder & Data Engineer
milos@thingsolver.com

Things Solver
ENLIGHTEN YOUR DATA