

# Simba: Efficient In-Memory Spatial Analytics.

Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou and Minyi Guo  
SIGMOD'16.

Andres Calderon

November 9, 2016

# Agenda

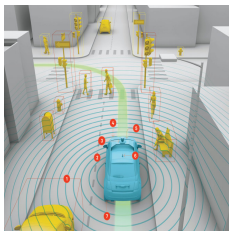
- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

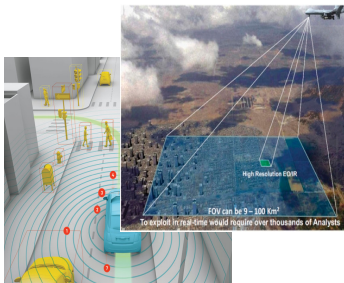
# Introduction

- There has been an explosion in the amount of spatial data in recent years...



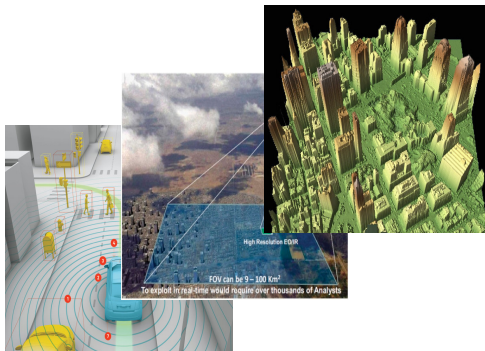
# Introduction

- There has been an explosion in the amount of spatial data in recent years...



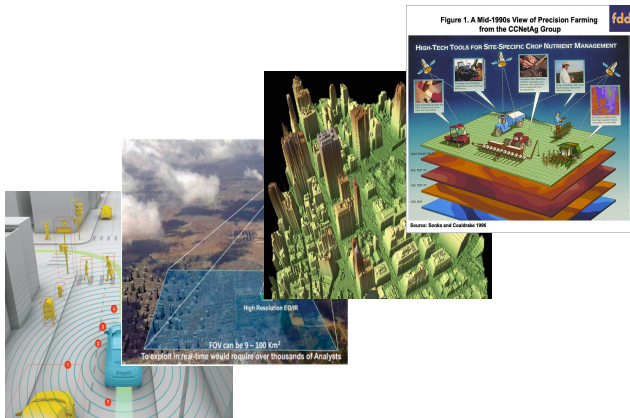
# Introduction

- There has been an explosion in the amount of spatial data in recent years...



# Introduction

- There has been an explosion in the amount of spatial data in recent years...



# Introduction

- The applications and commercial interest is clear...





# Introduction

- But remember that “Spatial is Special” ...



# Introduction

- But remember that “Spatial is Special” ...



ORACLE<sup>®</sup>  
SPATIAL



Hadoop-GIS  
*Spatial Big Data Solutions*



MD-Hbase



SECONDO



# Introduction

- But remember that “Spatial is Special” ...



ORACLE<sup>®</sup>  
SPATIAL



Hadoop-GIS  
*Spatial Big Data Solutions*



MD-Hbase



SECONDO



GeoSpark



GeoTrellis

Spark<sup>SQL</sup>

SpatialSpark



# Introduction

- Why do we need a new tool???



# Introduction

- Problems of Existing Systems...

- Single node database (low scalability)  
ArcGIS, PostGIS, Oracle Spatial.
- Disk-oriented cluster computation (low performance)  
Hadoop-GIS, SpatialHadoop, GeoMesa.
- No native support for spatial operators  
Spark SQL, MemSQL
- No sophisticated query planner and optimizer  
SpatialSpark, GeoSpark

# Introduction

- Simba: **S**patial **I**n **M**emory **B**ig data **A**alytics.
  - 1 Extends Spark SQL to support spatial queries and offers simple APIs for both SQL and DataFrame.
  - 2 Support two-layer spatial indexing over RDDs (low latency).
  - 3 Designs a SQL context to run important spatial operations in parallel (high throughput).
  - 4 Introduces spatial-aware and cost-based optimizations to select good spatial plans.

# Introduction

- Simba: **S**patial **I**n **M**emory **B**ig data **A**alytics.
  - 1 Extends Spark SQL to support spatial queries and offers simple APIs for both SQL and DataFrame.
  - 2 Support two-layer spatial indexing over RDDs (low latency).
  - 3 Designs a SQL context to run important spatial operations in parallel (high throughput).
  - 4 Introduces spatial-aware and cost-based optimizations to select good spatial plans.

# Introduction

- Simba: **S**patial **I**n **M**emory **B**ig data **A**alytics.
  - 1 Extends Spark SQL to support spatial queries and offers simple APIs for both SQL and DataFrame.
  - 2 Support two-layer spatial indexing over RDDs (low latency).
  - 3 Designs a SQL context to run important spatial operations in parallel (high throughput).
  - 4 Introduces spatial-aware and cost-based optimizations to select good spatial plans.



# Introduction

- Simba: **S**patial **I**n **M**emory **B**ig data **A**alytics.
  - 1 Extends Spark SQL to support spatial queries and offers simple APIs for both SQL and DataFrame.
  - 2 Support two-layer spatial indexing over RDDs (low latency).
  - 3 Designs a SQL context to run important spatial operations in parallel (high throughput).
  - 4 Introduces spatial-aware and cost-based optimizations to select good spatial plans.

# Introduction

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
<b>query operation support</b>					
Box range query	✓	✓	✓	✓	✓
Circle range query	✓	✓	✓	×	×
$k$ nearest neighbor	✓	✓	only 1NN	✓	×
Distance join	✓	✓	✓	via spatial join	✓
$k$ NN join	✓	×	×	×	×
Geometric object	×	✓	✓	✓	✓
Compound query	✓	×	×	✓	×

**Table 1: Comparing Simba against other systems.**

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Spark SQL Overview

Spark SQL is Apache Spark's module for working with structured data.

- Seamlessly mixes SQL queries with Spark programs.
- Connects to any data source the same way.
- Includes a highly extensible cost-based optimizer (*Catalyst*).
- Spark SQL is a full-fledged query engine based on the underlying Spark core.

# Spark SQL Overview

Spark SQL is Apache Spark's module for working with structured data.

- Seamlessly mixes SQL queries with Spark programs.
- Connects to any data source the same way.
- Includes a highly extensible cost-based optimizer (*Catalyst*).
- Spark SQL is a full-fledged query engine based on the underlying Spark core.

# Spark SQL Overview

Spark SQL is Apache Spark's module for working with structured data.

- Seamlessly mixes SQL queries with Spark programs.
- Connects to any data source the same way.
- Includes a highly extensible cost-based optimizer (*Catalyst*).
- Spark SQL is a full-fledged query engine based on the underlying Spark core.

# Spark SQL Overview

Spark SQL is Apache Spark's module for working with structured data.

- Seamlessly mixes SQL queries with Spark programs.
- Connects to any data source the same way.
- Includes a highly extensible cost-based optimizer (*Catalyst*).
- Spark SQL is a full-fledged query engine based on the underlying Spark core.

# Spark SQL Overview

Spark SQL is Apache Spark's module for working with structured data.

- Seamlessly mixes SQL queries with Spark programs.
- Connects to any data source the same way.
- Includes a highly extensible cost-based optimizer (*Catalyst*).
- Spark SQL is a full-fledged query engine based on the underlying Spark core.

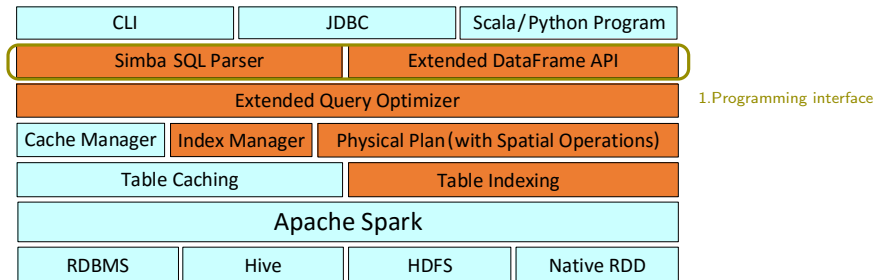


# Spark SQL Overview

```
# Apply functions to results of SQL queries.
context = HiveContext(sc)
results = context.sql("""
    SELECT
        *
    FROM
        people""")
names = results.map(lambda p: p.name)
# Query and join different data sources.
context.jsonFile("s3n://...").registerTempTable("json")
results = context.sql("""
    SELECT
        *
    FROM
        people
    JOIN
        json ...""")
```

# Simba Architecture

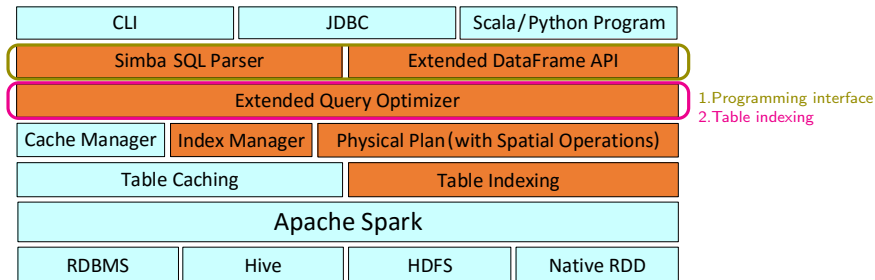
Simba is an extension of Spark SQL across the system stack.



**Figure 1: Simba architecture.**

# Simba Architecture

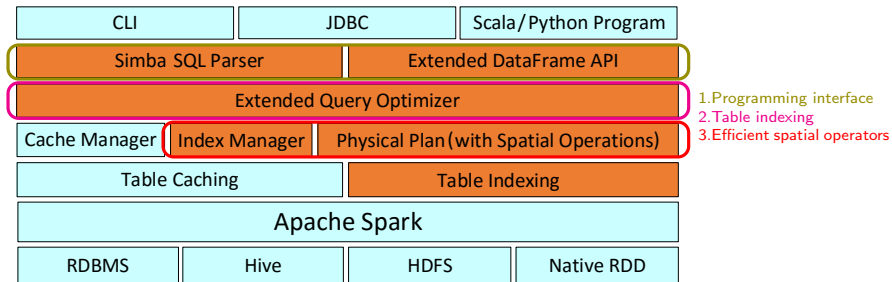
Simba is an extension of Spark SQL across the system stack.



**Figure 1: Simba architecture.**

# Simba Architecture

Simba is an extension of Spark SQL across the system stack.



**Figure 1: Simba architecture.**

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions



# Agenda

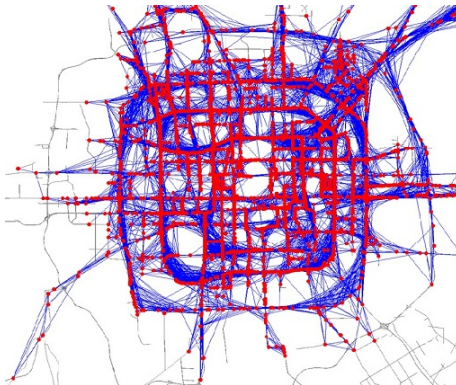
- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Setup

- Cluster of 25 nodes:
  - HDD from 50GB to 200GB.
  - RAM from 2GB to 8GB.
  - Processors 2.2GHz to 3GHz
- Single machine:
  - HDD 2TB.
  - RAM 16GB.
  - Processor 3.4GHz.

# Datasets

- Real datasets (from OpenStreetMap):
  - OSM1: 164M polygons, 80GB.
  - OSM2: 1.7B points, 52GB.
- Synthetic dataset:
  - SYNTH: 3.8B points, 128GB.
  - Five different distributions.



# Agenda

- 1 Background
- 2 Simba Architecture Overview
  - Programming Interface
  - Indexing
  - Spatial Operations
  - Optimization
- 3 Experiments
- 4 Conclusions

# Conclusions

- This paper introduced CG\_Hadoop as a scalable and efficient MapReduce library.
- Focused on 5 fundamental computational geometry problems...
  - Polygon union, Skyline, Convex hull, Farthest and Closest Pairs.
- Provided versions for Apache Hadoop and SpatialHadoop systems.
- Distributed approach speed up performance.
- Spatial partitioning allows early pruning which make it even more efficient.
- Achieve up to 29x and 260x better performance.

# Future ideas

- Working on more complex operations, for example motion patterns.
- Explore ports to new distributed platforms such as Spark or Simba.

# Thank you!!!

Do you have any question?